فصل

۱۰

استخراج و اجماع

مقدمه

عبارت «استخراج [معدنکاوی] بیت‌کوین» قدری گمراه‌کننده است. وقتی واژه‌ی استخراج (mining) را می‌شنویم، ناخودآگاه به یاد استخراج فلزات گرانبها از معدن می‌افتیم، و توجه ما به جایزه‌ای جلب می‌شود که برای این کار در نظر گرفته شده است، یعنی سکه‌های بیت‌کوین جدیدی که در هر بلاک تولید می‌شوند. اگر چه استخراج بیت‌کوین با این جایزه تشویق می‌شود، اما هدف اصلی فرآیند استخراج بیت‌کوین به دست آوردن این جایزه یا تولید سکه‌های جدید نیست. اگر به استخراج بیت‌کوین فقط به عنوان فرآیندی برای تولید سکه نگاه کنید، وسیله (ابزار تشویق) را با هدف این فرآیند اشتباه گرفته‌اید. «استخراج» ساز و کاری است که از اتاق پایاپای غیر متمرکز بیت‌کوین (جایی که تراکُنش‌ها اعتبارسنجی و تسویه می‌شوند) پشتیبانی می‌کند. فرآیند استخراج همان ایده‌ی ابتکاری است که بیت‌کوین را از سایر ارزهای رمزبنیان پیش از آن متمایز کرده است: یک ساز و کار امنیتی غیر متمرکز که اساس پول دیجیتال P2P را تشکیل می‌دهد.

فرآیند استخراج *امنیت سیستم بیت‌کوین را تأمین می‌کند* و به آن اجازه می‌دهد تا بتوان *بدون داشتن یک مرجع مرکزی به اجماع در سطح شبکه* دست یافت. جایزه‌ی سکه‌های تازه ضرب‌شده و کارمزد تراکُنش یک ساز و کار تشویقی برای همسو کردن فعالیت معدنچیان با امنیت شبکه است، و در عین حال روشی برای تأمین ذخایر پولی آن نیز محسوب می‌شود.

توجه هدف فرآیند استخراج تولید سکه‌ی بیت‌کوین جدید نیست؛ این فقط سیستم تشویقی آن است. استخراج (معدنکاوی) ساز و کاری است که امنیت بیت‌کوین را به صورت غیرمتمرکز درمی‌آورد.

معدنچیان بیت‌کوین تراکُنش‌های جدید را اعتبارسنجی کرده و آنها را در *دفتر کل جهانی* (بلاک‌چین) ثبت می‌کنند. بلاک جدید (که حاوی تراکُنش‌های رُخ‌داده بعد از آخرین بلاک است) به طور میانگین هر ۱۰ دقیقه «استخراج» می‌شود و آن تراکُنش‌ها را به بلاک‌چین اضافه می‌کند. قرار گرفتن تراکُنش‌ها در یک بلاک و اضافه شدن به بلاک‌چین به معنای «تأیید» آنها است، فرآیندی که به مالکان جدید آن بیت‌کوین‌ها اجازه می‌دهد آنها را در تراکُنش‌های بعدی خود خرج کنند.

از آنجا که فرآیند استخراج متضمن امنیت بیت‌کوین است، معدنچی (miner) به ازای این تلاش دو نوع جایزه (پاداش) دریافت می‌کند: سکه‌های جدیدی که در هر بلاک جدید تولید می‌شوند، و کارمزد تمام تراکُنش‌هایی که در آن بلاک قرار دارند.

برای تصاحب این جایزه، معدنچیان به رقابت برای حل یک مسأله‌ی ریاضی دشوار بر مبنای الگوریتم دَرهم‌سازی رمزنگاری می‌پردازند. حل این مسأله، که به آن اثبات-کار (Proof-of-Work) یا به اختصار PoW می‌گویند، به‌عنوان مدرک اثبات تلاش کامپیوتری مصرف‌شده توسط استخراج‌کننده (معدنچی) آن بلاک، در داخل بلاک جدید قرار داده می‌شود. رقابت برای حل الگوریتم اثبات-کار برای کسب جایزه‌ی بلاک و حق ثبت تراکُنش‌ها در بلاک‌چین اساس مدل امنیتی بیت‌کوین را تشکیل می‌دهد.

این فرآیند را استخراج یا معدنکاوی می‌نامند، چون جایزه‌ی آن (تولید سکه‌های جدید) به گونه‌ای طراحی شده که درست مثل استخراج فلزات قیمتی *بازده نزولی* داشته باشد. ذخایر پولی بیت‌کوین (مانند انتشار اسکناس توسط بانک‌های مرکزی) از طریق فرآیند استخراج افزایش می‌یابد. حداکثر مقدار بیت‌کوین جدید یا جایزه‌ای که معدنچیان می‌توانند با استخراج بلاک‌های جدید کسب کنند، تقریباً هر چهار سال (یا به طور دقیق، هر ۲۱۰,۰۰۰ بلاک) کاهش می‌یابد (نصف می‌شود). این جایزه از ۵۰ بیت‌کوین به ازای هر بلاک در ژانویه ۲۰۰۹ شروع شد و در نوامبر ۲۰۱۲ به ۲۵ بیت‌کوین به ازای هر بلاک کاهش یافت؛ از ژوئیه ۲۰۱۶ به این سو نیز مقدار این جایزه به ۱۲٫۵ بیت‌کوین به ازای استخراج هر بلاک تقلیل یافته است. بر اساس این فرمول، جایزه‌ی استخراج بیت‌کوین تا حدود سال ۲۱۴۰، یعنی وقتی تمام ذخیره‌ی بیت‌کوین (۲۰٫۹۹۹۹۹۹۹۸ میلیون بیت‌کوین) استخراج شود، به صورت نمایی کاهش خواهد یافت؛ بعد از سال ۲۱۴۰ دیگر هیچ بیت‌کوین جدیدی منتشر نخواهد شد.

معدنچیان بیت‌کوین از کارمزد تراکُنش‌ها نیز درآمد کسب می‌کنند. وقتی یک تراکُنش انجام می‌دهید، می‌توانید مقداری کارمزد (در قالب اختلاف ورودی‌ها و خروجی‌های تراکُنش) برای آن در نظر بگیرید. برنده‌ی مسابقه‌ی استخراج بیت‌کوین کارمزد تمامی تراکُنش‌های موجود در بلاک برنده را تصاحب خواهد کرد. امروزه فقط نیم درصد درآمد معدنچیان بیت‌کوین مربوط به کارمزد تراکُنش‌ها است؛ ۹۹٫۵ درصد درآمد این معدنچیان را استخراج بلاک‌های جدید و تصاحب جوایز آنها تشکیل می‌دهد. با این حال، مقدار این جایزه با گذشت زمان کاهش یافته و تعداد تراکُنش‌های هر بلاک افزایش خواهد یافت، که در نتیجه سهم کارمزد تراکُنش‌ها در درآمد معدنچیان بیت‌کوین سال به سال بیشتر خواهد شد. به تدریج این نسبت به نفع کارمزد تراکُنش تغییر خواهد کرد؛ و سرانجام، وقتی در سال ۲۱۴۰ جایزه‌ی استخراج بلاک‌های جدید به صفر برسد، کارمزد تراکُنش‌ها به تنها مشوق معدنچیان بیت‌کوین تبدیل خواهد شد.

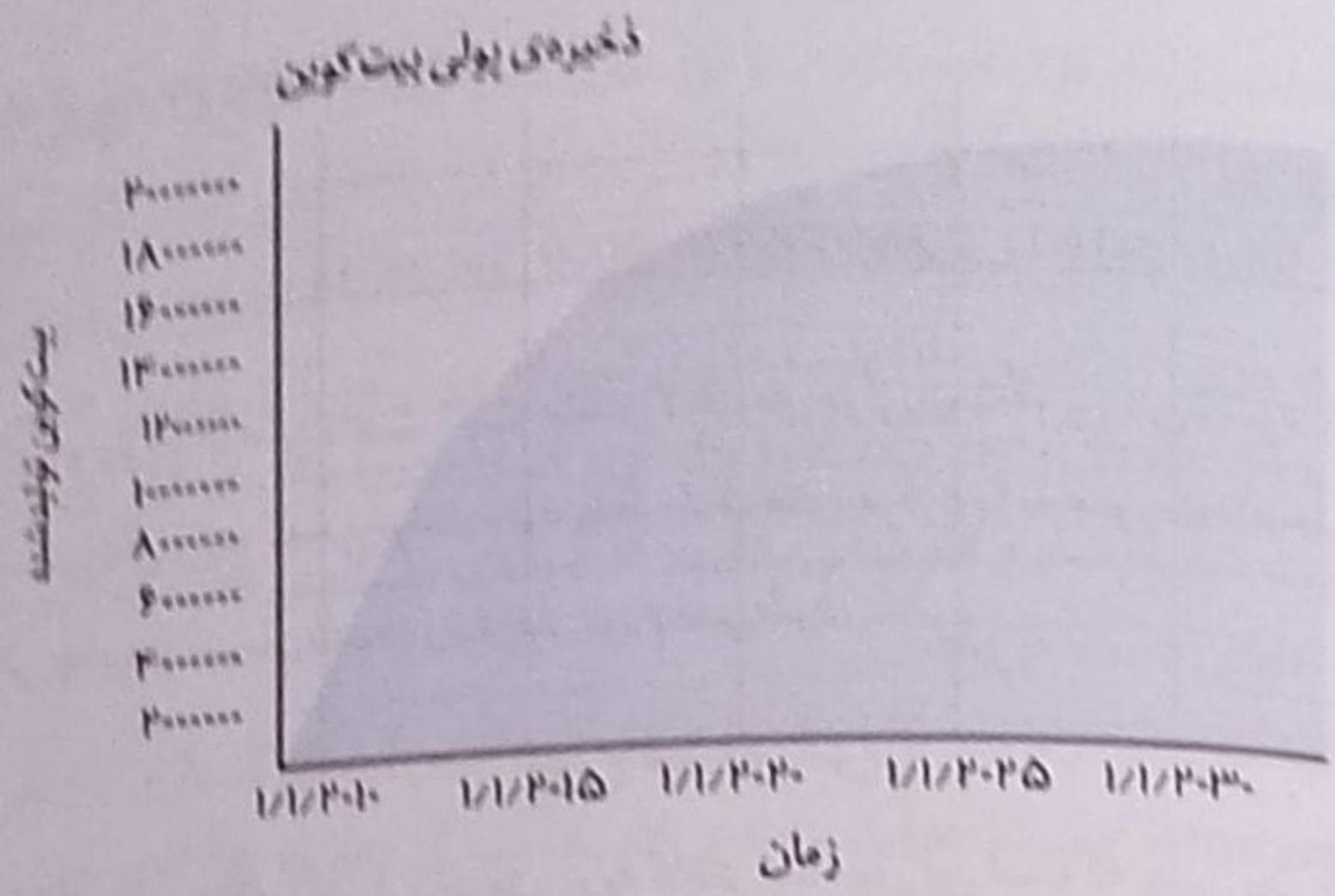
در این فصل ابتدا به بررسی فرآیند استخراج (معدنکاوی) به عنوان یک ساز و کار تولید پول می‌پردازیم و سپس نگاهی به مهم‌ترین کارکرد این فرآیند می‌اندازیم: ساز و کار اجماع غیرمتمرکز که اساس امنیت بیت‌کوین را تشکیل می‌دهد.

برای درک بهتر استخراج و اجماع از همان تراکُنش آلیس استفاده می‌کنیم، تراکُنشی که جینگ [دانشجوی اهل شانگهای که یک مزرعه‌ی استخراج بیت‌کوین دارد] آن را استخراج کرده و به بلاک‌چین اضافه می‌کند، و بدین ترتیب این بیت‌کوین از طریق فرآیندی موسوم به *اجماع برآیندی* در شبکه‌ی بیت‌کوین پذیرفته می‌شود.

اقتصاد بیت‌کوین و خلق پول

ضرب سکه‌های جدید بیت‌کوین و خلق پول جدید، با آهنگی ثابت و در حال نزول، در فرآیند تولید بلاک‌ها صورت می‌گیرد. هر بلاک، که به طور میانگین در هر ۱۰ دقیقه تولید می‌شود، حاوی سکه‌های جدید بیت‌کوین است که از هیچ خلق شده‌اند. هر ۲۱۰,۰۰۰ بلاک، یا تقریباً هر چهار سال، آهنگ انتشار پول جدید نصف می‌شود.

در نوامبر ۲۰۱۲، آهنگ انتشار بیت‌کوین جدید (که در چهار سال ابتدایی ظهور بیت‌کوین ۵۰ بیت‌کوین بر بلاک بود) به ۲۵ بیت‌کوین به ازای هر بلاک کاهش یافت، و در ژوئیه ۲۰۱۶ به ۱۲٫۵ بیت‌کوین بر بلاک رسید. این آهنگ در بلاک ۶۳۰,۰۰۰ که در اوایل یا اواسط سال ۲۰۲۰ استخراج خواهد شد، به ۶٫۲۵ بیت‌کوین بر بلاک تقلیل می‌یابد. بدین ترتیب، بعد از گذشت ۳۲ دوره‌ی چهار ساله، یعنی در بلاک ۶,۷۲۰,۰۰۰ (که استخراج آن حدود سال ۲۱۳۷ اتفاق خواهد افتاد)، جایزه‌ی استخراج بلاک جدید به کوچکترین واحد سیستم پولی بیت‌کوین، ۱ ساتوشی، کاهش خواهد یافت. سرانجام، بعد از بلاک ۶,۹۳۰,۰۰۰ (در حدود سال ۲۱۴۰)، مقدار پول عرضه‌شده در سیستم بیت‌کوین به تقریباً ۲,۰۹۹,۹۹۹,۹۹۷,۶۹۰,۰۰۰ ساتوشی (حدود ۲۱ میلیون بیت‌کوین) خواهد رسید.



شکل ۱۰-۱ آهنگ عرضه‌ی پول جدید بیت‌کوین در طی زمان به صورت نمایی کاهش می‌یابد.

از آن به بعد، بلاک‌های جدید حاوی بیت‌کوین جدید نخواهند بود، و کارمزد تراکنش‌ها تنها جایزه‌ی فرآیند استخراج را تشکیل خواهد داد. نمودار مقدار پول در گردش در سیستم بیت‌کوین در طول زمان را در شکل ۱۰-۱ مشاهده می‌کنید.

نکته حداکثر تعداد بیت‌کوین‌های قابل استخراج (۲۱ میلیون بیت‌کوین) کران بالا (سقف) جایزه‌ی در نظر گرفته شده برای فرآیند استخراج بیت‌کوین است. در عمل، ممکن است یک معدنچی به طور عمد جایزه‌ی کامل استخراج یک بلاک را تصاحب نکند؛ این قبیل بلاک‌ها قبلاً استخراج شده‌اند و شاید در آینده هم استخراج شوند. در نتیجه، مجموع کل سکه‌های بیت‌کوین منتشرشده عملاً کمتر از این سقف خواهد بود.

در مثال ۱۰-۱ یک اسکریپت پایتون ساده به نام *max_money.py* می‌بینید که مقدار کل بیت‌کوینی را که منتشر خواهد شد، محاسبه می‌کند؛ مثال ۱۰-۲ نیز خروجی این اسکریپت را نشان می‌دهد.

مثال ۱۰-۱ برنامه‌ای برای محاسبه‌ی مقدار کل بیت‌کوینی که منتشر خواهد شد

```
# Original block reward for miners was 50 BTC
start_block_reward = 50
# 210000 is around every 4 years with a 10 minute block interval
reward_interval = 210000

def max_money():
    # 50 BTC = 50 0000 0000 Satoshis
    current_reward = 50 * 10**8
    total = 0
    while current_reward > 0:
        total += reward_interval * current_reward
        current_reward /= 2
    return total

print "Total BTC to ever be created:", max_money(), "Satoshis"
```

مثال ۱۰-۲ اجرای اسکریپت max_money.py

```
$ python max_money.py
Total BTC to ever be created: 2099999997690000 Satoshis
```

انتشـــار محدود و رو به کاهش باعث ثابت ماندن ذخیره‌ی پولی بیت‌کوین می‌شـــود که اثر ضدتورمی دارد. بر خلاف ارزهای بی‌پشـــتوانه‌ی رایج که می‌توانند (توســـط بانک‌های مرکزی) به صورت نامحدود چاپ و منتشر شوند، بیت‌کوین هرگز در اثر چاپ و انتشار نامحدود دچار تورم نمی‌شود.

پول ضدتورمی

مهم‌ترین و مجادله‌برانگیزترین پیامد انتشـــار محدود و رو به کاهش یک پول این است که چنین پولی ذاتاً ضدتورمی خواهد بود. به پدیده‌ی افزایش ارزش یک پول در اثر عدم توازن بین عرضه و تقاضا (و بالا رفتن نرخ تبدیل آن) تورم کاهی گفته می‌شـود. بر خلاف پول‌های تورمی، قدرت خرید یک پول ضدتورمی در طول زمان افزایش می‌یابد.

بسیاری از اقتصاددانان معتقدند اقتصاد ضدتورمی یک فاجعه است که باید به هر قیمتی از آن اجتناب کرد، چون در چنین اقتصادی مردم به امید ســقوط بیشــتر قیمت کالاها، به جای خرج کردن پول به انباشتن آن تمایل می‌یابند. چنین پدیده‌ای در دهه‌ی ۱۹۹۰ در ژاپن رُخ داد؛ در این دوره‌ی رکود اقتصادی (که به «دهه‌ی گمشده» معروف شد)، افت شدید تقاضا در بازار باعث شد ارزش پول ملی ژاپن (ین) سیر صعودی پیش بگیرد و باعث تعمیق هر چه بیشتر رکود شود.

از ســوی دیگر، متخصصان بیت‌کوین اعتقاد دارند تورم‌کاهی در ذات خود پدیده‌ی بدی نیســت. ما تورم‌کاهی را برای اقتصاد مضر می‌دانیم چون تورم‌کاهی ناشــی از ســقوط تقاضا تنها نمونه از این پدیده بوده اســت که با آن برخورد داشته‌ایم. در اقتصادهای متکی به ارزهای بی‌پشــتوانه که امکان چاپ و نشر نامحدود پول وجود دارد، بدون سقوط تقاضا و عدم تمایل دولت به چاپ پول، بروز پدیده‌ی تورم‌کاهی بسیار دشوار [و عملاً غیرممکن] است. تورم‌کاهی بیت‌کوین ناشی از سقوط تقاضا نیست، بلکه نتیجه‌ی یک محدودیت قابل‌پیش‌بینی در عرضه‌ی پول است.

البته، جنبه‌ی مثبت تورم‌کاهی همان ضد تورم بودن آن است. تورم به طور تدریجی، ولی اجتناب‌ناپذیر، ارزش پول را کاهش می‌دهد و با تنبیه پنهان کسانی که پول خود را پس‌انداز می‌کنند، در نهایت به نفع وام‌گیرندگان، به ویژه وام‌گیرندگان بزرگ [و در رأس آنها، خود دولت‌ها]، عمل می‌کند. ارزهایی که تحت کنترل دولت‌ها هستند، همیشه در معرض خطر پرداخت وام‌ها و تسهیلات آسان قرار دارند، وام‌هایی که بعداً می‌توان آنها را از طریق کاهش ارزش پول (و به هزینه‌ی پس‌اندازکنندگان) پاک کرد.

آینده نشان خواهد داد که وقتی جنبه‌ی ضدتورمی یک پول ناشی از سقوط سریع اقتصاد نباشد، آیا مشکل‌ساز خواهد بود یا (با جلوگیری از تورم و کاهش ارزش پول) یک مزیت محسوب می‌شود.

اجماع غیرمتمرکز

در فصل قبل بلاک‌چین بیت‌کوین را بررسی کردیم، و گفتیم بلاک‌چین عبارت است از دفتر کل (فهرست) جهانی تمامی تراکُنش‌هایی که در شـــبکه‌ی بیت‌کوین توسط همگان پذیرفته شده‌اند، و همین پذیرش همگانی نشانه‌ی صلاحیت ثبت مالکیت آنها است. ولی در یک شبکه‌ی جهانی که هیچ کس دیگری را نمی‌شناسد و به آنها اعتماد ندارد، توافق عام بر سر یک «حقیقت» جهان‌شمول واحد که چه چیزی متعلق به چه کسی است، چگونه اتفاق می‌افتد؟

تمامی نظام‌های پرداخت سنتی وابسته به یک «مدل اعتماد» هستند، مدلی که در آن یک نهاد مسئول مرکزی (عمدتاً) از طریق اعتبارســـنجی و تسویه‌ی تراکُنش‌های مالی، عملیات پایاپای (تسویه‌حساب بین افراد و بانک‌ها) را انجام می‌دهد.

بیت‌کوین دارای یک نهاد مسئول مرکزی نیست، با این حال، تمام گره‌های کامل یک کپی کامل از دفتر کل را که می‌توانند (به عنوان مدرک معتبر) به آن اعتماد کنند، در اختیار دارند. این بلاک‌چین (دفتر کل جهانی) توسط یک نهاد مرکزی ایجاد نشده، بلکه به طور مستقل توسط تک تک گره‌های شبکه ساخته و تکمیل می‌شود.

همه‌ی گره‌های شبکه که روی اطلاعات مبادله‌شده روی شبکه‌ای با اتصالات ناایمن کار می‌کنند، می‌توانند (از طریقی که خواهید دید) به یک نتیجه‌ی واحد برسند و یک کپی کاملاً مشابه از دفتر کلی که نزد همگان وجود دارد، داشته باشند. در این فصل به تشریح ساز و کاری می‌پردازیم که شبکه‌ی بیت‌کوین به کمک آن و بدون داشتن یک نهاد مسئول مرکزی به اجماع جهانی دست می‌یابد.

اختراع اصلی ساتوشی ناکاموتو [در پی‌ریزی سیستم بیت‌کوین] همان ساز و کار غیرمتمرکز برای اجماع برآیندی (emergent consensus) است. صفت *برآیندی* اشاره به آن دارد که این اجماع به طور صریح حاصل نمی‌شود: هیچ شورا یا انتخاباتی در کار نیست و هیچ زمان مشخصی را نمی‌توان به عنوان لحظه‌ی ظهور اجماع تعیین کرد. به جای آن، اجماع برآیندی محصول برهم‌کُنش غیرهمزمان هزاران گره مستقل است، گره‌هایی که همگی چند قاعده‌ی ساده را دنبال می‌کنند. تمام خصوصیات بیت‌کوین که به یک نهاد مرکزی یا مدل اعتماد وابسته نیستند (از جمله پول بیت‌کوین، تراکُنش‌ها، پرداخت‌ها، و مدل امنیت آن)، از همین ابداع سرچشمه می‌گیرند.

اجماع غیرمتمرکز بیت‌کوین برآیند برهم‌کُنش چهار فرآیند است که به طور مستقل در گره‌های سرتاسر شبکه روی می‌دهند:

- اعتبارسنجی مستقل هر تراکُنش، توسط تمام گره‌های کامل، بر اساس فهرستی جامع از معیارها؛
- تجمیع مستقل این تراکُنش‌ها در بلاک‌های جدید توسط گره‌های معدنچی، در کنار ارائه‌ی مدرک برای پردازش انجام‌شده از طریق الگوریتم اثبات-کار؛
- اعتبارسنجی مستقل بلاک‌های جدید در تمام گره‌ها و اضافه کردن آنها به زنجیره‌ی بلاک‌ها (بلاک‌چین)؛
- انتخاب مستقل زنجیره‌ای با بیشترین پردازش انباشته از طریق الگوریتم اثبات-کار.

در چند قسمت آینده این فرآیندها را بررسی کرده و نشان می‌دهیم که برهم‌کُنش آنها چگونه به یک اجماع برآیندی در سرتاسر شبکه منجر می‌شود، اجماعی که به تک تک گره‌های بیت‌کوین اجازه می‌دهد یک کپی کامل، قابل اعتماد، قابل استناد، و عمومی از دفتر کل جهانی (بلاک‌چین) در اختیار داشته باشند.

اعتبارسنجی مستقل تراکُنش‌ها

در فصل ۶ دیدیم که نرم‌افزار کیف‌پول تولید یک تراکُنش را با جمع‌آوری تعدادی UTXO شروع کرده، و با ایجاد اسکریپت بازکننده‌ی قفل مناسب، و سپس ساخت خروجی(های) جدید در وجه مالک جدید (دریافت‌کننده) به پایان می‌رساند. این تراکُنش به گره‌های همسایه (همتا) در شبکه‌ی بیت‌کوین فرستاده می‌شود تا به تدریج در سرتاسر شبکه‌ی بیت‌کوین منتشر شود.

با این حال، هر گره بیت‌کوین که تراکُنشی را دریافت می‌کند، قبل از ارسال آن به همسایگان خود، ابتدا این تراکُنش را اعتبارسنجی خواهد کرد. این کار تضمین می‌کند که فقط تراکُنش‌های معتبر روی شبکه‌ی بیت‌کوین منتشر شده، و تراکُنش‌های نامعتبر در همان اولین گره دور انداخته شوند. گره‌های بیت‌کوین اعتبار یک تراکُنش را به کمک فهرست طویلی از معیارهای از پیش تعیین‌شده ارزیابی می‌کنند:

- ساختار دستوری و ساختمان داده‌ی تراکُنش باید درست باشد.
- هیچ‌یک از ورودی‌ها یا خروجی‌های آن نباید خالی باشند.
- اندازه‌ی تراکُنش بر حسب بایت نباید از MAX_BLOCK_SIZE بیشتر باشد.

- مقدار هر خروجی و مجموع کل آنها باید در محدوده‌ی مجاز (کمتر از ۲۱ میلیون بیت‌کوین، و بیشتر از آستانه‌ی پشیز) باشد.
- درهم هیچ از ورودی‌ها نباید ۰ باشد؛ و همچنین مقدار N نباید ۱- باشد (تراکُنش‌های پایگاه‌سکه نباید منتشر شوند).
- مقدار فیلد nLocktime باید برابر با INT_MAX باشد، یا مقادیر nLocktime و nSequence باید معیار MedianTimePast را برآورده کنند.
- اندازه‌ی تراکُنش بر حسب بایت باید بزرگتر یا مساوی ۱۰۰ باشد.
- تعداد عمل‌های امضا (SIGOPS) موجود در تراکنش باید کوچکتر از محدوده‌ی مجاز *عمل امضا* باشد.
- اسکریپت بازکننده‌ی قفل (scriptSig) فقط می‌توانند اعداد را به داخل پشته بفرستد، و اسکریپت قفل‌کننده (scriptPubKey) باید با اَشکال تعریف‌شده با عملگر isStandard منطبق باشد (این معیار باعث مردود شدن تراکُنش‌های «غیراستاندارد» می‌شود).
- تراکُنش‌های اشاره‌شده باید در مخزن تراکُنش، یا در یک بلاک در شاخه‌ی اصلی، وجود داشته باشند.
- برای هر ورودی، اگر خروجی ارجاع‌شده در دیگر تراکُنش‌های مخزن تراکُنش وجود داشته باشد، آن تراکُنش باید مردود شود.
- برای هر ورودی، تراکُنش خروجی ارجاع‌شده بایستی در شاخه‌ی اصلی و مخزن تراکُنش جستجو شود. اگر این تراکُنش خروجی (برای هر ورودی) یافت نشود، آن تراکُنش یک تراکُنش یتیم تلقی خواهد شد. اگر چنین تراکُنشی از قبل در مخزن تراکُنش‌های یتیم وجود نداشته باشد، در این مخزن قرار داده می‌شود.
- برای هر ورودی، اگر تراکُنش خروجی ارجاع‌شده یک خروجی پایگاه‌سکه باشد، بایستی حداقل COINBASE_MATURITY (در حال حاضر ۱۰۰) تأییدیه داشته باشد.
- برای هر ورودی، خروجی ارجاع‌شده بایستی وجود داشته و قبلاً خرج نشده باشد.
- مقدار هر ورودی و همچنین جمع کل آنها بایستی با استفاده از تراکُنش‌های خروجی ارجاع‌شده در هر ورودی بررسی شود؛ این مقادیر باید در محدوده‌ی مجاز (کمتر از ۲۱ میلیون بیت‌کوین، بیشتر از ۰) قرار داشته باشند.
- تراکُنشی که مجموع مقدار ورودی‌های آن کمتر از جمع مقدار خروجی‌های تراکُنش باشد، مردود می‌شود.
- اگر کارمزد تراکُنش کمتر از حد معین برای قرار گرفتن در یک بلاک خالی (minRelayTxFee) باشد، مردود می‌شود.
- اسکریپت بازکننده‌ی قفل هر ورودی باید با اسکریپت قفل‌کننده‌ی خروجی متناظر اعتبارسنجی شود.

این شروط را می‌توان در توابع AcceptToMemoryPool، CheckTransaction و CheckInputs در هسته‌ی بیت‌کوین به تفصیل مشاهده کرد. توجه داشته باشید که این شروط در طول زمان عوض می‌شوند تا با حمله‌های جدید به سیستم بیت‌کوین مقابله شود؛ و گاهی برخی از آنها تعدیل می‌شوند تا امکان استفاده از انواع بیشتری از تراکُنش‌ها فراهم شود.

هر گره بیت‌کوین با اعتبارسنجی مستقل تراکُنش‌هایی که دریافت می‌کند (و قبل از انتشار بیشتر آنها در شبکه)، یک مخزن از تراکُنش‌های معتبر (ولی تأییدنشده) می‌سازد که به آن **مخزن تراکُنش** (transaction pool) یا **مخزن حافظه** (memory pool) گفته می‌شود.

گره استخراج (معدنکاوی)

برخی گره‌ها در شبکه‌ی بیت‌کوین فقط یک وظیفه‌ی تخصصی یعنی استخراج بیت‌کوین بر عهده دارند، و به همین دلیل به آنها گره معدنچی (miner) گفته می‌شود. در فصل ۱ با جینگ دانشجوی مهندسی کامپیوتر در شانگهای (چین) آشنا شدیم که معدنچی بیت‌کوین است. جینگ برای به دست آوردن سکه‌های جدید از «تجهیزات استخراج» استفاده می‌کند، نوعی سیستم‌های سخت‌افزاری که به طور خاص برای استخراج بیت‌کوین طراحی شده‌اند. مجموعه‌ی این تجهیزات سخت‌افزاری (که به آن *مزرعه‌ی استخراج* نیز گفته می‌شود) به یک سرویس‌دهنده متصل شده‌اند که در واقع یک گره کامل بیت‌کوین است. بر خلاف جینگ، برخی از دیگر معدنچیان عملیات استخراج را بدون داشتن یک گره کامل انجام می‌دهند که در ادامه‌ی همین فصل روش کار آنها را توضیح خواهیم داد. گره جینگ، درست مثل هر گره کامل دیگر، تراکُنش‌های تأییدنشده را دریافت کرده و آنها را روی شبکه‌ی بیت‌کوین بازنشر می‌کند، ولی کار دیگری نیز انجام می‌دهد: تجمیع این تراکُنش‌ها در یک بلاک جدید.

گره جینگ (مانند همه‌ی گره‌ها) بلاک‌های جدید را نیز دریافت کرده و آنها را روی شبکه‌ی بیت‌کوین بازنشر می‌کند. با این حال، ورود یک بلاک جدید اهمیت خاصی برای یک گره معدنچی دارد. وقتی گره معدنچی یک بلاک جدید دریافت می‌کند، یعنی رقابت بر سر آن بلاک به پایان رسیده و برنده (کسی که جایزه‌ی استخراج بلاک جدید را دریافت می‌کند) مشخص شده است. ورود یک بلاک معتبر جدید برای گره‌های معدنچی به معنای شکست در این رقابت است. ولی پایان هر دور رقابت به معنای شروع دور جدید نیز هست. به عبارت دیگر، انتشار یک بلاک معتبر جدید در شبکه‌ی بیت‌کوین همزمان به معنای سوت پایان یک مسابقه و سوت آغاز مسابقه‌ی بعدی است.

تجمیع مستقل تراکُنش‌ها در بلاک

بعد از اعتبارسنجی یک تراکُنش، گره بیت‌کوین آن را به مخزن حافظه (یا مخزن تراکُنش) اضافه می‌کند، جایی که تراکُنش‌ها تا زمان قرار گرفتن در یک بلاک (استخراج شدن) در آنجا منتظر می‌مانند. گره جینگ تراکُنش‌های جدید را جمع‌آوری، اعتبارسنجی، و (درست مثل گره‌های دیگر) در شبکه منتشر خواهد کرد. با این حال، بر خلاف گره‌های دیگر، گره جینگ این تراکُنش‌ها را در بلاکی موسوم به **بلاک نامزد** (candidate block) تجمیع نیز می‌کند.

اجازه دهید فرآیند ایجاد بلاک را از لحظه‌ی خرید یک فنجان قهوه توسط آلیس از مغازه‌ی باب دنبال کنیم. در فصل‌های قبل دیدیم که تراکُنش آلیس در بلاک ۲۷۷۳۱۶ قرار داده شد. برای اهداف آموزشی این فصل فرض می‌کنیم این بلاک توسط مزرعه‌ی استخراج جینگ استخراج شده است و تراکُنش آلیس را بعد از قرار گرفتن در این بلاک جدید دنبال خواهیم کرد.

گره استخراج جینگ یک گره کامل است، بنابراین یک نسخه‌ی کامل و به‌روز از بلاک‌چین در اختیار دارد. وقتی آلیس فنجان قهوه را از مغازه‌ی باب می‌خرد، گره جینگ زنجیره‌ی بلاک‌چین را تا بلاک ۲۷۷۳۱۴ مونتاژ کرده است. این گره با گوش کردن (دریافت) تراکُنش‌های منتشرشده روی شبکه تلاش می‌کند یک بلاک جدید استخراج کند، و در همان حال به بلاک‌هایی که توسط گره‌های دیگر کشف شده‌اند، گوش می‌کند. در حالی که گره جینگ مشغول استخراج بلاک جدید است، بلاک ۲۷۷۳۱۵ را از طریق شبکه‌ی بیت‌کوین دریافت می‌کند؛ ورود این بلاک به معنای آن است که رقابت بر سر بلاک ۲۷۷۳۱۵ به پایان رسیده و رقابت برای استخراج بلاک ۲۷۷۳۱۶ آغاز شده است.

در طول همان ۱۰ دقیقه‌ای که گره جینگ مشغول محاسبه برای حل معمای رمزنگاری بلاک ۲۷۷۳۱۵ است، به جمع‌آوری تراکُنش‌ها و آماده شدن برای بلاک بعدی نیز ادامه می‌دهد، به طوری که در لحظه‌ی ورود بلاک ۲۷۷۳۱۵، این گره چند صد تراکُنش در مخزن حافظه‌ی خود جمع‌آوری کرده است. با دریافت این بلاک و در همان

حال که مشغول اعتبارسنجی آن است، گره جینگ تراکُنش‌های موجود در مخزن حافظه‌ی خود را با تراکُنش‌های بلاک ۲۷۷۳۱۵ مقایسه کرده و آن تراکُنش‌هایی را که در این بلاک وجود دارند، از مخزن حافظه‌ی خود حذف می‌کند. تراکُنش‌های باقیمانده در مخزن حافظه‌ی گره جینگ آنهایی هستند که هنوز تأیید نشده‌اند و منتظر ثبت شدن در یک بلاک جدید هستند.

گره جینگ بلافاصله یک بلاک خالی جدید ایجاد می‌کند که یک گزینه‌ی احتمالی برای بلاک ۲۷۷۳۱۶ است. به این بلاک *بلاک نامزد* گفته می‌شود چون هنوز یک بلاک معتبر نیست و اثبات-کار معتبر ندارد. این بلاک زمانی فقط معتبر می‌شود که معدنچی بتواند یک حل برای معمای رمزنگاری الگوریتم اثبات-کار آن بیابد.

وقتی گره جینگ تمام تراکُنش‌های تأییدنشده در مخزن حافظه‌ی خود را تجمیع می‌کند، یک بلاک نامزد جدید با ۴۱۸ تراکُنش و کارمزد تراکُنش کل ۰/۰۹۰۹۴۹۲۸ بیت‌کوین به دست می‌آورد. به کمک فرمان getblockhash رابط کاربری مشتری هسته‌ی بیت‌کوین می‌توانید این بلاک را در بلاک‌چین مشاهده کنید. مثال ۱۰-۳ را ببینید.

مثال ۱۰-۳ استفاده از خط-فرمان مشتری هسته‌ی بیت‌کوین برای بازیابی بلاک ۲۷۷۳۱۶

```
$ bitcoin-cli getblockhash 277316
0000000000000001b6b9a13b095e96db41c4a928b97ef2d944a9b31b2cc7bdc4

$ bitcoin-cli getblock 0000000000000001b6b9a13b095e96db41c4a928b97ef2d944a9b31b
2cc7bdc4

{
  "hash" : "0000000000000001b6b9a13b095e96db41c4a928b97ef2d944a9b31b2cc7bdc4",
  "confirmations" : 35561,
  "size" : 218629,
  "height" : 277316,
  "version" : 2,
  "merkleroot" :
    "c91c008c26e50763e9f548bb8b2fc323735f73577effbc55502c51eb4cc7cf2e",
  "tx" : [
    "d5ada064c6417ca25c4308bd158c34b77e1c0eca2a73cda16c737e7424afba2f",
    "b268b45c59b39d759614757718b9918caf0ba9d97c56f3b91956ff877c503fbe",

    ... 417 more transactions ...

    ],
  "time" : 1388185914,
  "nonce" : 924591752,
  "bits" : "1903a30c",
  "difficulty" : 1180923195.25802612,
  "chainwork" :
    "0000000000000000000000000000000000000000000934695e92aaf53afa1a",
  "previousblockhash" :
    "0000000000000002a7bbd25a417c0374cc55261021e8a9ca74442b01284f0569"
}
```

تراکُنش پایگاه‌سکه

اولین تراکُنش هر بلاک یک تراکُنش خاص موسوم به تراکُنش پایگاه‌سکه (coinbase transaction) است. این تراکُنش توسط گره معدنچی (در اینجا، گره جینگ) ساخته می‌شود و حاوی *جایزه‌ی استخراج بلاک* (یا *جایزه‌ی پایگاه‌سکه*) است.

در زمان استخراج بلاک ۲۷۷۳۱۶، جایزه‌ی استخراج هر بلاک ۲۵ بیت‌کوین بود. همان طور که گفتیم، جایزه‌ی استخراج بلاک در بازه‌های ۴-ساله (یا به بیان دقیق‌تر، هر ۲۱۰۰۰۰ بلاک) نصف می‌شود، و این جایزه اکنون (از ژوییه ۲۰۱۶ به این طرف) به ۱۲٫۵ بیت‌کوین کاهش یافته است. نصف شدن بعدی در سال ۲۰۲۰ خواهد بود.

گره جینگ تراکُنش پایگاه‌سکه را به عنوان یک پرداخت به کیف‌پول خودش ایجاد می‌کند: «۲۵٫۰۹۰۹۴۹۲۸ بیت‌کوین به آدرس جینگ پرداخت کن.» همان طور که متوجه شده‌اید، این مبلغ برابر است با مجموع جایزه‌ی پایگاه‌سکه (۲۵ بیت‌کوین جدید) و کارمزد تمامی تراکُنش‌های موجود در این بلاک (۰٫۰۹۰۹۴۹۲۸ بیت‌کوین)؛ مثال ۱۰-۴ را ببینید.

مثال ۱۰-۴ تراکُنش پایگاه‌سکه

```
$ bitcoin-cli getrawtransaction
d5ada064c6417ca25c4308bd158c34b77e1c0eca2a73cda16c737e7424afba2f 1

{
    "hex" :
"010000000100000000000000000000000000000000000000000000000000000000000000ffff
ffff0f03443b0403858402062f503253482ffffffff0110c08d9500000000232102aa970c59264
0d19de03ff6f329d6fd2eecb023263b9ba5d1b81c29b523da8b21ac00000000",
    "txid" : "d5ada064c6417ca25c4308bd158c34b77e1c0eca2a73cda16c737e7424afba2f",
    "version" : 1,
    "locktime" : 0,
    "vin" : [
        {
            "coinbase" : "03443b0403858402062f503253482f",
            "sequence" : 4294967295
        }
    ],
    "vout" : [
        {
            "value" : 25.09094928,
            "n" : 0,
            "scriptPubKey" : {
                "asm" :
"02aa970c592640d19de03ff6f329d6fd2eecb023263b9ba5d1b81c29b523da8b21OP_CHECKSIG"
                "hex" :
"2102aa970c592640d19de03ff6f329d6fd2eecb023263b9ba5d1b81c29b523da8b21ac",
                "reqSigs" : 1,
```

```
                "type" : "pubkey",
                "addresses" : [
                    "1MxTkeEP2PmHSMze5tUZ1hAV3YTKu2Gh1N"
                ]
            }
        }
    ]
}
```

بر خلاف تراکُنش‌های معمولی، تراکُنش پایگاه‌سکه UTXO ورودی ندارد؛ به جای آن، این تراکنش یک ورودی به نام پایگاه‌سکه (coinbase) دارد که از هیچ بیت‌کوین خلق می‌کند. تراکُنش پایگاه‌سکه یک خروجی دارد که کل مبلغ تراکُنش را به آدرس بیت‌کوین مالک گره (کیف‌پول) پرداخت می‌کند. در این تراکُنش پایگاه‌سکه مبلغ ۲۵٫۰۹۰۹۴۹۲۸ بیت‌کوین به آدرس کیف‌پول جینگ (1MxTkeEP2PmHSMze5tUZ1hAV3YTKu2Gh1N) پرداخت شده است.

محاسبه‌ی جایزه پایگاه‌سکه و کارمزد تراکُنش‌های بلاک

برای ایجاد تراکُنش پایگاه‌سکه، گره جینگ ابتدا با استفاده از فرمول زیر کارمزد ۴۱۸ تراکُنش موجود در بلاک جدید را محاسبه می‌کند؛ همان طور که دیدیم، این کارمزد برای بلاک ۲۷۷۳۱۶ معادل ۰٫۰۹۰۹۴۹۲۸ بیت‌کوین است:

```
Total Fees = Sum(Inputs) - Sum(Outputs)
```

پس از آن، گره جینگ جایزه‌ی استخراج بلاک جدید را محاسبه می‌کند. جایزه‌ی پایگاه‌سکه به ارتفاع بلاک بستگی دارد، و از ۵۰ بیت‌کوین به ازای استخراج هر بلاک شروع شده و با اضافه شدن هر ۲۱۰,۰۰۰ بلاک نصف می‌شود. از آنجا که ارتفاع این بلاک ۲۷۷۳۱۶ است، جایزه‌ی استخراج آن ۲۵ بیت‌کوین خواهد بود. در مشتری هسته‌ی بیت‌کوین تابعی به نام GetBlockSubsidy وجود دارد که برای محاسبه‌ی جایزه‌ی پایگاه‌سکه به کار می‌رود؛ این تابع را در مثال ۱۰-۵ مشاهده می‌کنید.

مثال ۱۰-۵ محاسبه‌ی جایزه‌ی پایگاه‌سکه: تابع GetBlockSubsidy() در ماژول main.cpp

```
CAmount GetBlockSubsidy(int nHeight, const Consensus::Params& consensusParams)
{
    int halvings = nHeight / consensusParams.nSubsidyHalvingInterval;
    // Force block reward to zero when right shift is undefined.
    if (halvings >= 64)
        return 0;

    CAmount nSubsidy = 50 * COIN;
    // Subsidy is cut in half every 210,000 blocks which will occur approximately
       every 4 years.
    nSubsidy >>= halvings;
    return nSubsidy;
}
```

این تابع ابتدا تعداد دفعات نصف‌شدن را از تقسیم صحیح ارتفاع بلاک فعلی بر ۲۱۰,۰۰۰ (بازه‌ی نصف شدن جایزه) به دست آورده و در متغیر halvings ذخیره می‌کند. خارج قسمت صحیح تقسیم ۲۷۷۳۱۶ بر ۲۱۰,۰۰۰ معادل ۱ است، بنابراین

جایزه‌ی پایگاه‌سکه باید یک بار نصف شود. متغیر nSubsidy نیز مقدار جایزه‌ی پایگاه‌سکه اولیه (۵۰ بیت‌کوین) را بر حسب ساتوشی نشان می‌دهد، که از ضرب ثابت COIN (که معادل ۱۰۰ میلیون ساتوشی = ۱ بیت‌کوین تعریف شده) در ۵۰ به دست می‌آید. تابع GetBlockSubsidy تقسیم بر ۲ را با عملگر انتقال-به-راست-باینری (>>) انجام می‌دهد، چون سرعت اجرای این عملگر بسیار بیشتر از عملگر تقسیم معمولی است. برای بلاک ۲۷۷۳۱۶، یک بار انتقال-به-راست-باینری (نصف کردن) جایزه‌ی پایگاه‌سکه را از ۵ میلیارد ساتوشی (۵۰ بیت‌کوین) به ۲٫۵ میلیارد ساتوشی (۲۵ بیت‌کوین) تبدیل می‌کند. از آنجا که متغیر nSubsidy یک عدد صحیح ۶۴-بیتی است، (برای جلوگیری از بروز هر گونه باگ) اگر تعداد انتقال-به-راست-باینری بزرگتر یا مساوی ۶۴ باشد، این تابع مقدار ۰ بر خواهد گرداند. در آخر، مقدار nSubsidy با کارمزد تراکنش‌های بلاک (nFees) جمع شده و به عنوان مبلغ خروجی تراکنش پایگاه‌سکه استفاده می‌شود.

توجه

اگر تراکنش پایگاه‌سکه را گره معدنچی تولید می‌کند، چه چیزی مانع آن می‌شود تا به خودش ۱۰۰ یا ۱۰۰۰ بیت‌کوین «جایزه» ندهد؟ پاسخ این است که جایزه‌ای با مقدار نادرست باعث می‌شود تا گره‌های دیگر آن را به عنوان یک بلاک نامعتبر شناسایی کنند، و تمام زحمتی که گره معدنچی برای استخراج آن کشیده (انرژی و زمانی که صرف یافتن حل معمای اثبات-کار کرده)، هدر برود. جایزه‌ی استخراج یک بلاک را فقط زمانی می‌توان خرج کرد که همه‌ی گره‌های شبکه‌ی بیت‌کوین آن را تأیید کرده باشند.

ساختار تراکُنش پایگاه‌سکه

بعد از انجام این محاسبات، گره جینگ باید یک تراکُنش پایگاه‌سکه برای پرداخت ۲۵٫۰۹۰۹۴۹۲۸ بیت‌کوین به خودش ایجاد کند. همان طور که در مثال ۱۰-۴ می‌بینید، تراکُنش پایگاه‌سکه فرمت خاصی دارد: این تراکُنش به جای خرج کردن یک UTXO موجود به عنوان ورودی، از یک ورودی «پایگاه‌سکه» استفاده می‌کند. ساختار ورودی‌های تراکُنش را در جدول ۶-۲ بررسی کردیم. اجازه دهید یک ورودی تراکُنش معمولی را با ورودی تراکُنش پایگاه‌سکه مقایسه کنیم. ساختار یک ورودی تراکُنش معمولی را در جدول ۱۰-۱، و ساختار یک ورودی تراکُنش پایگاه‌سکه را در جدول ۱۰-۲ مشاهده می‌کنید.

جدول ۱۰-۱ ساختار ورودی تراکُنش «معمولی»

فیلد	اندازه	توضیح
درهم تراکُنش	۳۲ بایت	ارجاع به تراکُنش حاوی UTXO که باید خرج شود.
اندیس خروجی	۴ بایت	اندیس UTXO که باید خرج شود (شماره‌گذاری از ۰).
اندازه‌ی اسکریپت بازکننده‌ی قفل	۱ تا ۹ بایت (متغیر)	طول اسکریپت بازکننده‌ی قفل (برحسب بایت).
اسکریپت بازکننده‌ی قفل	متغیر	اسکریپتی که شرایط خرج کردن UTXO را برآورده می‌کند.
شماره‌ترتیب	۴ بایت	فعلاً غیرفعال؛ مقدار 0xFFFFFFFF می‌گیرد.

جدول ۱۰-۲ ساختار ورودی تراکُنش «پایگاه‌سکه»

فیلد	اندازه	توضیح
درهم تراکُنش	۳۲ بایت	تمام بیت‌ها «صفر»؛ تراکُنش ورودی وجود ندارد.
اندیس خروجی	۴ بایت	تمام بیت‌ها «یک»: 0xFFFFFFFF.
اندازه‌ی داده‌ی پایگاه‌سکه	۱ تا ۹ بایت (متغیر)	طول داده‌ی پایگاه‌سکه، از ۲ تا ۱۰۰ بایت.
داده‌ی پایگاه‌سکه	متغیر	داده‌های اختیاری برای نونس اضافی و برچسب‌های استخراج. در بلاک‌های v2 باید با ارتفاع بلاک آغاز شود.
شماره‌ترتیب	۴ بایت	مقدار 0xFFFFFFFF می‌گیرد.

در تراکُنش پایگاه‌سکه، دو فیلد اول (دَرهم تراکُنشی که باید خرج شود، و اندیس UTXO مورداستفاده در این تراکُنش) مقادیری می‌گیرند که به هیچ UTXO یا تراکُنشی ارجاع نمی‌کنند: در این تراکُنش به تمام بیت‌های فیلد اول (دَرهم تراکُنش) مقدار 0، و به تمام بیت‌های فیلد دوم (اندیس خروجی) مقدار 1 داده می‌شود. در فیلد «اسکریپت بازکننده‌ی قفل» (scriptSig) هم به جای یک اسکریپت واقعی، داده‌هایی نوشته می‌شود که فقط برای معدنچی کاربرد دارند.

داده‌ی پایگاه‌سکه

از آنجا که تراکُنش‌های پایگاه‌سکه UTXO ورودی ندارند، و نیازی هم به اسکریپت بازکننده‌ی قفل (scriptSig) ندارند، این فیلد جای خود را به «داده‌ی پایگاه‌سکه» داده است که باید بین ۲ تا ۱۰۰ بایت باشد. به جز چند بایت اول، بقیه‌ی بایت‌های این فیلد می‌توانند هر چیزی باشند؛ در واقع، داده‌های موجود در این فیلد کاملاً اختیاری است.

برای مثال، ساتوشی ناکاموتو در فیلد داده‌ی تراکُنش پایگاه‌سکه‌ی بلاک زاینده عبارت «روزنامه‌ی تایمز، ۳ ژانویه ۲۰۰۹: نخست‌وزیر در آستانه‌ی دومین استیضاح به دلیل بحران بانکی» را نوشت تا مدرکی باشد برای اثبات تاریخ ایجاد این تراکُنش. امروزه، اغلب معدنچیان از این فیلد برای پیوست کردن نونس‌های اضافی و همچنین معرفی مزرعه‌ی استخراج خود استفاده می‌کنند.

در ابتدا تمامی بایت‌های فیلد «داده‌ی پایگاه‌سکه» اختیاری بودند، ولی امروزه دیگر چنین نیست. بنا بر BIP-34، بلاک‌های ویرایش-۲ (بلاک‌هایی که فیلد version آنها مقدار 2 دارد) بایستی به عنوان یک عمل «به داخل پشته فرستادن» اسکریپت در ابتدای فیلد پایگاه‌سکه، حاوی اندیس ارتفاع بلاک باشند. در مورد بلاک ۲۷۷۳۱۶ دیدیم که فیلد اسکریپت بازکننده‌ی قفل یا scriptSig ورودی تراکُنش [پایگاه‌سکه] مقدار 03443b0403858402062f503253482f دارد (مثال ۱۰-۴ را ببینید). اجازه دهید ببینیم این اعداد چه معنایی دارند.

بایت اول، 03، به موتور اجرای اسکریپت دستور می‌دهد تا ۳ بایت بعدی را به داخل پشته‌ی اسکریپت بفرستد. سه بایت بعدی، 0x443b04، ارتفاع بلاک هستند که به صورت راست-نویس (little-endian: فرمتی که در آن بایت‌های یک رشته از راست به چپ در خروجی قرار داده می‌شوند و ترتیب آنها معکوس می‌شود) کُدگذاری شده‌اند. اگر ترتیب این سه بایت را معکوس کنید، به عدد هگزادسیمال 0x043b44 می‌رسید که معادل ۲۷۷۳۱۶ دهدهی است.

پنج بایت بعدی، 0385840206، برای کُدگذاری یک نونس (nonce) یا رشته‌ی *تصادفی* اضافی به کار می‌رود که کاربرد آن دریافتن یک راه‌حل مناسب برای معمای رمزنگاری اثبات-کار است (برای آشنایی با مفهوم نونس ادامه‌ی همین فصل را ببینید).

آخرین بخش از داده‌ی پایگاه‌سکه (2f503253482f) رشته‌ی متنی /P2SH/ با کُدگذاری ASCII است، که می‌گوید گره استخراج‌کننده‌ی این بلاک از پرداخت P2SH (که در استاندارد BIP-16 تعریف شده) پشتیبانی می‌کند. بعد از معرفی روش پرداخت P2SH، گره‌های معدنچی باید به طور صریح اعلام می‌کردند از کدام یک از استانداردهای BIP-16 یا BIP-17 پشتیبانی می‌کنند. گره‌هایی که از پیاده‌سازی BIP-16 پشتیبانی می‌کردند، باید عبارت /P2SH/ را در فیلد داده‌ی پایگاه‌سکه خود می‌نوشتند؛ و گره‌هایی که از پیاده‌سازی BIP-17 پشتیبانی می‌کردند، بایستی عبارت p2sh/CHV را در این فیلد قرار می‌دادند. سرانجام، استاندارد BIP-16 به عنوان برنده‌ی این رقابت انتخاب شد و دیگر نیازی به این تفکیک نبود، ولی بسیاری از گره‌های معدنچی همچنان به درج عبارت /P2SH/ در فیلد داده‌ی پایگاه‌سکه خود ادامه دادند (و می‌دهند).

در مثال ۱۰-۶ یک برنامه‌ی C++ کوچک به نام satoshi-words می‌بینید که با استفاده از توابع کتابخانه‌ی libbitcoin (که در فصل ۳ معرفی کردیم) پیام معروف ساتوشی ناکاموتو را از فیلد داده‌ی پایگاه‌سکه‌ی بلاک زاینده خوانده و نمایش

می‌دهد. توجه کنید که کتابخانه‌ی libbitcoin حاوی یک کپی ثابت از بلاک زاینده است، بنابراین برنامه‌ی زیر به طور مستقیم به این بلاک دسترسی دارد و نیازی به دسترسی به بلاک‌چین نیست.

مثال ۱۰-۶ خواندن محتویات فیلد داده‌ی پایگاه‌سکه از بلاک زاینده

```
/*
  Display the genesis block message by Satoshi.
*/
#include <iostream>
#include <bitcoin/bitcoin.hpp>

int main()
{
    // Create genesis block.
    bc::chain::block block = bc::chain::block::genesis_mainnet();
    // Genesis block contains a single coinbase transaction.
    assert(block.transactions().size() == 1);
    // Get first transaction in block (coinbase).
    const bc::chain::transaction& coinbase_tx = block.transactions()[0];
    // Coinbase tx has a single input.
    assert(coinbase_tx.inputs().size() == 1);
    const bc::chain::input& coinbase_input = coinbase_tx.inputs()[0];
    // Convert the input script to its raw format.
    const auto prefix = false;
    const bc::data_chunk& raw_message = coinbase_input.script().to_data(prefix);
    // Convert this to a std::string.
    std::string message(raw_message.begin(), raw_message.end());
    // Display the genesis block message.
    std::cout << message << std::endl;
    return 0;
}
```

اگر این برنامه را به صورتی که در مثال ۱۰-۷ می‌بینید، با کامپایلر GNU C++ کامپایل کرده و سپس اجرا کنید، پیام ساتوشی ناکاموتو در بلاک زاینده را مشاهده خواهید کرد.

مثال ۱۰-۷ نمایش پیام ساتوشی ناکاموتو در بلاک زاینده

```
$ # Compile the code
$ g++ -o satoshi-words satoshi-words.cpp $(pkg-config --cflags --libs libbitcoin)
$ # Run the executable
$ ./satoshi-words
^D^^<GS>^A^DEThe Times 03/Jan/2009 Chancellor on brink of second bailout for
banks
```

ایجاد سرآیند بلاک نامزد

برای ساخت سرآیند بلاک نامزد، گره معدنچی باید شش فیلد را پُر کند؛ این فیلدها را در جدول ۱۰-۳ مشاهده می‌کنید.

جدول ۱۰-۳ ساختار سرآیند بلاک نامزد

فیلد	اندازه	توضیح
ویرایش	۴ بایت	عددی که شماره‌ی ارتقای پروتکل/نرم‌افزار را مشخص می‌کند.
دَرهم بلاک قبلی	۳۲ بایت	ارجاع به دَرهم بلاک قبلی (والد) در زنجیره.
ریشه‌ی مرکل	۳۲ بایت	دَرهم ریشه‌ی درخت مرکل تراکُنش‌های این بلاک.
برچسب زمان	۴ بایت	زمان تقریبی ایجاد این بلاک (ثانیه‌های سپری‌شده از نیمه‌شب اول ژانویه ۱۹۷۰).
هدف	۴ بایت	هدف الگوریتم اثبات-کار این بلاک.
نونس (رشته‌ی تصادفی)	۴ بایت	یک شمارنده که برای الگوریتم اثبات-کار استفاده می‌شود.

در زمان استخراج بلاک ۲۷۷۳۱۶، دومین ویرایش ساختار بلاک معرفی شده بود؛ این عدد با فرمت راست-نویس به صورت 0x02000000 در فیلد ۴-بایتی «ویرایش» نوشته می‌شود. گره معدنچی باید در فیلد دوم، «دَرهم بلاک قبلی» (prevhash)، دَرهم سرآیند بلاک *مادر* این بلاک (در لحظه‌ی شروع استخراج بلاک نامزد)، یعنی سرآیند بلاک ۲۷۷۳۱۵، را قرار دهد؛ این دَرهم 0000000000000002a7bbd25a417c0374cc55261021e8a9ca74442b01284f0569 است (مثال ۱۰-۳ را ببینید).

نکته

با انتخاب یک بلاک مادر (والد) خاص و قرار دادن آن در فیلد «دَرهم بلاک قبلی» بلاک نامزد، گره جینگ قدرت پردازش خود را در خدمت توسعه‌ی زنجیره‌ای قرار می‌دهد که به آن بلاک ختم شده است. در حقیقت، جینگ با قدرت پردازش خود به این زنجیره به عنوان معتبرترین زنجیره «رأی» می‌دهد.

در گام بعدی، گره معدنچی باید تمامی تراکُنش‌های این بلاک را در یک درخت مرکل فشرده کرده و ریشه‌ی این درخت را در فیلد سوم سرآیند بلاک نامزد، فیلد «ریشه‌ی مرکل»، ثبت کند. اولین تراکُنشی که در درخت مرکل قرار می‌گیرد، تراکُنش پایگاه‌سکه است، و پس از آن ۴۱۸ تراکُنش موجود در مخزن حافظه به درخت مرکل اضافه می‌شوند که تعداد کل گره‌های این درخت را به ۴۱۹ می‌رساند. همان طور که می‌دانید، تعداد گره‌های «برگ» در یک درخت مرکل باید زوج باشد، بنابراین آخرین تراکُنش تکرار می‌شود تا تعداد گره‌ها به ۴۲۰ برسد. با فرآیندی که در فصل قبل تشریح کردیم، این گره‌ها دو به دو با یکدیگر دَرهم‌سازی می‌شوند و سطوح جدید ایجاد می‌کنند، تا سرانجام فقط یک گره (ریشه‌ی مرکل) باقی بماند. ریشه‌ی مرکل یک مقدار ۳۲-بایتی واحد است؛ همان طور که در مثال ۱۰-۳ می‌بینید، ریشه‌ی مرکل بلاک ۲۷۷۳۱۶ چنین است:

c91c008c26e50763e9f548bb8b2fc323735f73577effbc55502c51eb4cc7cf2e

گره جینگ سپس زمان (تقریبی) ایجاد این بلاک نامزد را بر حسب تعداد ثانیه‌های سپری‌شده از نیمه‌شب اول ژانویه ۱۹۷۰ (به وقت گرینویچ) در فیلد ۴-بایتی «برچسب زمان» قرار می‌دهد. مقدار 1388185914 که در مثال ۱۰-۳ می‌بینید، معادل جمعه، ۲۷ دسامبر ۲۰۱۳، ساعت ۲۳:۱۱:۵۴ (GMT) است.

فیلد بعدی که گره جینگ باید پُر کند، فیلد هدف (target) است که اثبات-کار مورد نیاز برای تبدیل این بلاک به یک بلاک معتبر را تعریف می‌کند. مقدار این اثبات-کار با فرمت «بیت‌های هدف» در این فیلد ذخیره می‌شود، که هدف [دشواری] الگوریتم اثبات-کار را به صورت یک عدد ۴-بایتی با کُدگذاری پایه-توان، ۱ بایت *توان* و ۳ بایت *پایه* (ضریب)، بیان می‌کند.

برای مثال، هدف [دشواری] بلاک ۲۷۷۳۱۶ با مقدار هگزادسیمال 0x1903a30c مشخص شده است: بخش 0x19 (معادل ۲۵ دهدهی) توان است، و بخش 0x03a30c (معادل ۲۳۸,۳۴۸ دهدهی) ضریب را تشکیل می‌دهد.

در ادامه‌ی همین فصل درباره‌ی مفاهیم «هدف» و «بیت‌های هدف» به طور مفصل صحبت خواهیم کرد.

فیلد آخر «نونس» (رشته‌ی تصادفی) است، که در ابتدا به آن مقدار ۰ داده می‌شود.

با پُر شدن همه‌ی فیلدها، اکنون سرآیند بلاک نامزد کامل است و فرآیند استخراج آن می‌تواند آغاز شود. وظیفه‌ی فرآیند استخراج یافتن یک مقدار برای نونس است به طوری که دَرهم سرآیند یک بلاک کوچکتر از هدف مشخص‌شده باشد. گره معدنچی بایستی میلیون‌ها تریلیون مقدار (تصادفی) را آزمایش کند تا مقداری که این شرط را برآورده می‌کند، بیابد.

استخراج بلاک نامزد

بعد از ایجاد بلاک نامزد توسط گره معدنچی (در اینجا، گره جینگ)، وقتی آن است که استخراج این بلاک توسط سخت‌افزار گره جینگ آغاز شود؛ همان طور که گفتیم، وظیفه‌ی فرآیند استخراج یافتن پاسخی برای معمای رمزنگاری الگوریتم اثبات-کار و معتبر کردن این بلاک است. در فصل‌های قبل تعداد زیادی تابع دَرهم‌سازی رمزنگاری دیدیم که در بخش‌های مختلف سیستم بیت‌کوین کاربرد دارند؛ در فرآیند استخراج بیت‌کوین از تابع دَرهم‌سازی SHA256 استفاده می‌شود.

در ساده‌ترین بیان، فرآیند استخراج چیزی نیست جز دَرهم‌سازی سرآیند بلاک، تغییر دادن یک پارامتر، تکرار دَرهم‌سازی، و ادامه‌ی این کار تا پیدا شدن دَرهمی که هدف [دشواری] تعیین‌شده را برآورده کند. خروجی تابع دَرهم‌سازی را نمی‌توان از قبل تعیین کرد؛ همچنین ایجاد الگویی که بتواند یک مقدار دَرهم مشخص تولید کند، نیز غیرممکن است. این ویژگی توابع دَرهم‌سازی بدان معنا است که برای یافتن یک هدف معین هیچ راهی جز آزمایش یک مقدار ورودی، تغییر تصادفی ورودی، و تکرار آزمایش وجود ندارد. همان طور که می‌بینید، یافتن دَرهم مطلوب فقط [و فقط] به شانس بستگی دارد.

الگوریتم اثبات-کار (PoW)

الگوریتم دَرهم‌سازی یک داده‌ی ورودی با طول-دلخواه می‌گیرد و یک خروجی با طول-ثابت قطعی (deterministic) تولید می‌کند که می‌توان آن را *اثرانگشت* ورودی دانست. صفت *قطعی* به این معنا است که به ازای هر ورودی معین همیشه یک خروجی یکسان تولید خواهد شد که بستگی به پیاده‌سازی الگوریتم دَرهم‌سازی ندارد. ویژگی کلیدی یک الگوریتم دَرهم‌سازی رمزنگاری در این است که یافتن دو ورودی متفاوت که یک اثرانگشت یکسان تولید کنند (اتفاقی که به آن *تصادم* گفته می‌شود)، از نظر محاسباتی ناشدنی است. در نتیجه، به غیر از آزمایش ورودی‌های تصادفی، تقریباً غیرممکن است بتوان یک ورودی را به گونه‌ای انتخاب کرد که یک اثرانگشت دلخواه ایجاد کند.

خروجی الگوریتم دَرهم‌سازی SHA256 (صرفنظر از اندازه‌ی ورودی) همیشه ۲۵۶ بیت [۳۲ بایت] طول دارد. در مثال ۱۰-۸ از پایتون برای محاسبه‌ی دَرهم SHA256 عبارت «I am Satoshi Nakamoto» استفاده کرده‌ایم.

مثال ۱۰-۸ مثالی از اجرای الگوریتم SHA256

```
$ python

Python 2.7.1
>>> import hashlib
>>> print hashlib.sha256("I am Satoshi Nakamoto").hexdigest()
5d7c7ba21cbbcd75d14800b100252d5b428e5b1213d27c385bc141ca6b47989e
```

در این مثال عبارت «I am Satoshi Nakamoto» به یک عدد ۳۲-بایتی (۲۵۶-بیتی) دَرهم‌سازی شده است. این عدد بر اساس تمامی اجزای عبارت ورودی ساخته شده است، به طوری که اضافه کردن حتی یک نقطه یا تغییر دادن حتی یک حرف ورودی، آن را به کلی عوض خواهد کرد.

برای دیدن تأثیر تغییر ورودی بر خروجی الگوریتم دَرهم‌سازی، اجازه دهید از یک برنامه‌ی کوچک پایتون کمک بگیریم. این برنامه که آن را در مثال ۱۰-۹ می‌بینید، هر بار یک کاراکتر (نونس) به انتهای عبارت «I am Satoshi Nakamoto» اضافه کرده و آن را با همان الگوریتم SHA256 دَرهم‌سازی می‌کند.

مثال ۱۰-۹ تغییر دادن خروجی الگوریتم SHA256 با اضافه کردن یک نونس

```
# example of iterating a nonce in a hashing algorithm's input

import hashlib

text = "I am Satoshi Nakamoto"

# iterate nonce from 0 to 19

for nonce in xrange(20):

    # add the nonce to the end of the text
    input = text + str(nonce)

    # calculate the SHA-256 hash of the input (text+nonce)
    hash = hashlib.sha256(input).hexdigest()

    # show the input and hash result
    print input, '=>', hash
```

اگر این اسکریپت پایتون را اجرا کنید، ۲۰ دَرهم متفاوت برای ۲۰ رشته‌ی ورودی که حاصل اضافه کردن یک عدد (۰ تا ۱۹) به انتهای این عبارت هستند، خواهید داشت؛ مثال ۱۰-۱۰ را ببینید.

مثال ۱۰-۱۰ تغییر دادن خروجی الگوریتم SHA256 با اضافه کردن یک نونس

```
$ python hash_example.py

I am Satoshi Nakamoto0 => a80a81401765c8eddee25df36728d732...
I am Satoshi Nakamoto1 => f7bc9a6304a4647bb41241a677b5345f...
I am Satoshi Nakamoto2 => ea758a8134b115298a1583ffb80ae629...
I am Satoshi Nakamoto3 => bfa9779618ff072c903d773de30c99bd...
I am Satoshi Nakamoto4 => bce8564de9a83c18c31944a66bde992f...
I am Satoshi Nakamoto5 => eb362c3cf3479be0a97a20163589038e...
I am Satoshi Nakamoto6 => 4a2fd48e3be420d0d28e202360cfbaba...
I am Satoshi Nakamoto7 => 790b5a1349a5f2b909bf74d0d166b17a...
I am Satoshi Nakamoto8 => 702c45e5b15aa54b625d68dd947f1597...
I am Satoshi Nakamoto9 => 7007cf7dd40f5e933cd89fff5b791ff0...
I am Satoshi Nakamoto10 => c2f38c81992f4614206a21537bd634a...
I am Satoshi Nakamoto11 => 7045da6ed8a914690f087690e1e8d66...
I am Satoshi Nakamoto12 => 60f01db30c1a0d4cbce2b4b22e88b9b...
```

```
I am Satoshi Nakamoto13 => 0ebc56d59a34f5082aaef3d66b37a66...
I am Satoshi Nakamoto14 => 27ead1ca85da66981fd9da01a8c6816...
I am Satoshi Nakamoto15 => 394809fb809c5f83ce97ab554a2812c...
I am Satoshi Nakamoto16 => 8fa4992219df33f50834465d3047429...
I am Satoshi Nakamoto17 => dca9b8b4f8d8e1521fa4eaa46f4f0cd...
I am Satoshi Nakamoto18 => 9989a401b2a3a318b01e9ca9a22b0f3...
I am Satoshi Nakamoto19 => cda56022ecb5b67b2bc93a2d764e75f...
```

همان طور که دیده می‌شود، هر عبارت یک دَرهم کاملاً متفاوت که کاملاً هم تصادفی به نظر می‌رسند، تولید کرده است. جالب اینجاست که اگر این برنامه را روی هر کامپیوتری اجرا کنید، دقیقاً همین خروجی‌ها را تولید خواهد کرد. مقداری که در این برنامه به عنوان بخش متغیر رشته‌ی ورودی انتخاب کرده‌ایم، نونس (nonce) نامیده می‌شود. کاربرد اصلی نونس تغییر دادن خروجی تابع رمزنگاری (در این مثال، الگوریتم SHA256) است.

برای آن که با این الگوریتم یک چالش (معمای رمزنگاری) بسازیم، ابتدا لازم است یک «هدف» انتخاب کنیم؛ برای مثال: عبارتی را بیابید که دَرهم هگزادسیمال آن با صفر شروع شود. خوشبختانه، این مسأله‌ی چندان سختی نیست: یک نگاه به خروجی برنامه‌ی hash_example.py (مثال ۱۰-۱۰) نشان می‌دهد که عبارت «I am Satoshi Nakamoto13» دَرهم 0ebc56d59a34f5082aaef3d66b37a661696c2b618e62432727216ba9531041a5 تولید کرده است که خواسته‌ی ما را برآورده می‌کند. در واقع، برای حل این معما به فقط ۱۳ آزمایش نیاز داریم. از نظر قوانین احتمالات، اگر خروجی تابع دَرهم‌سازی توزیع یکنواخت داشته باشد، از هر ۱۶ خروجی یکی با 0 شروع خواهد شد (چون دستگاه عددنویسی هگزادسیمال ۱۶ رقم [0 تا F] دارد). از نظر عددی، این چالش معادل آن است که بگوییم عبارتی را بیابید که مقدار دَرهم آن از 0x1000000000000000000000000000000000000000000000000000000000000000 کوچکتر باشد. این آستانه را هدف می‌نامیم و برنامه‌ای می‌نویسیم که چنین دَرهمی را بیابد. توجه داشته باشید که هر چه این هدف کوچکتر انتخاب شود، یافتن دَرهمی که از آن کوچکتر باشد، دشوارتر خواهد شد.

برای درک بهتر این موضوع، یک بازی ساده را در نظر بگیرید که با دو تاس انجام می‌شود و هدف آن پرتابی است که مجموع خال‌های دو تاس از یک مقدار معین کوچکتر باشد. در دور اول، هدف را ۱۲ انتخاب می‌کنیم. بازنده شدن در این بازی فقط در یک حالت ممکن است: بازیکن جفت شش بیاورد. از آنجا که پرتاب همزمان دو تاس ۳۶ ترکیب مختلف ایجاد می‌کند و فقط یک ترکیب خاص بازنده است، احتمال برنده شدن در این بازی ۳۵ به ۳۶ (بیش از ۹۷٫۲ درصد) خواهد بود. در دور دوم، هدف را ۱۱ انتخاب می‌کنیم. اکنون مجموع خال‌های بازیکن باید کمتر از ۱۱ باشد تا برنده شود، و همچنان شانس زیادی برای برنده شدن خواهد داشت. اگر همین روند را ادامه دهیم، چند دور بعد هدف به ۵ کاهش می‌یابد، و شانس برنده شدن بازیکن به کمتر از نصف خواهد رسید. سرانجام، با کاهش هدف به ۲ (کمترین مقدار ممکن)، بازیکن فقط در یک حالت می‌تواند برنده شود: جفت یک بیاورد، که احتمال آن ۱ به ۳۶ (کمتر از ۲٫۸ درصد) است. همان طور که می‌بینید، با کوچکتر شدن هدف، احتمال برنده شدن در این بازی کم و کمتر می‌شود.

کسی که به این بازی نگاه می‌کند و برنده شدن چند بازیکن را زمانی که هدف ۲ است، می‌بیند، می‌تواند فرض را بر آن بگذارد که این بازیکنان به طور میانگین ۳۶ بار تاس انداخته‌اند. به عبارت دیگر، با دانستن «هدف» می‌توان سختی [مقدار تلاش مورد نیاز برای] برنده شدن در یک چالش [بازی] را تخمین زد. وقتی الگوریتم [چالش] بر اساس یک تابع دَرهم‌سازی مانند SHA256 بنا شده باشد، خود این خروجی *اثبات می‌کند* که چه مقدار *کار* برای تولید یک دَرهم کوچکتر از آن «هدف» صرف شده است؛ و به همین دلیل است که به آن اثبات-کار (Proof-of-Work)، یا به اختصار **PoW**، گفته می‌شود.

توجه اگر چه هر بار اجرای الگوریتم دَرهم‌سازی SHA256 یک خروجی تصادفی تولید می‌کند، اما احتمال تولید هر خروجی را می‌توان از پیش محاسبه کرد. بنابراین، وقتی برای هدفی با دشواری مشخص یک خروجی تولید می‌کنید، خود آن خروجی اثباتی است برای صرف مقدار کار تعیین‌شده.

در مثال ۱۰-۱۰، نونس برنده ۱۳ است و این نتیجه را هر کسی به طور مستقل می‌تواند تأیید کند. در واقع، هر کسی می‌تواند عدد ۱۳ را به انتهای عبارت «I am Satoshi Nakamoto» اضافه کرده و بعد از محاسبه‌ی دَرهم آن با الگوریتم SHA256، متوجه شود که این دَرهم از «هدف» تعیین‌شده کوچکتر است. خود همین دَرهم یک اثبات-کار نیز هست، چون ثابت می‌کند که کار لازم برای محاسبه‌ی آن را انجام داده‌ایم. هر چند برای یافتن این نونس لازم بود ۱۳ دَرهم را محاسبه کنیم، ولی برای تأیید آن فقط به یک محاسبه نیاز است. اگر هدف را کوچکتر (با سطح دشواری بالاتر) انتخاب کرده بودیم، برای یافتن نونس مناسب باید دَرهم‌های بیشتر را محاسبه می‌کردیم، ولی در هر حال تأیید این نونس (همیشه) فقط به یک محاسبه نیاز دارد. علاوه بر آن، با دانستن «هدف»، می‌توان با استفاده از تکنیک‌های آماری دشواری رسیدن به این هدف و از آنجا کار لازم برای یافتن نونس مناسب را تخمین زد.

توجه الگوریتم اثبات-کار باید دَرهمی تولید کند که کوچکتر از هدف تعیین‌شده باشد. هر چه این هدف بزرگتر باشد، دشواری یافتن دَرهم کوچکتر از آن کمتر است. از طرف دیگر، یک هدف کوچک به معنای بودن دشواری بیشتر در یافتن دَرهم مناسب خواهد بود. به بیان دقیق، «هدف» و «دشواری» رابطه‌ای معکوس دارند.

الگوریتم اثبات-کار بیت‌کوین بسیار شبیه چالشی است که در مثال ۱۰-۱۰ دیدیم. [گره] معدنچی یک بلاک نامزد می‌سازد و آن را با تعدادی تراکُنش پُر می‌کند؛ سپس، دَرهم سرآیند این بلاک را محاسبه کرده و آن را با هدف فعلی [شبکه‌ی بیت‌کوین] مقایسه می‌کند. اگر دَرهم محاسبه‌شده کوچکتر از این هدف نباشد، معدنچی نونس را تغییر می‌دهد (معمولاً فقط با یک واحد افزایش)، و دوباره آن را امتحان می‌کند. با سطح دشواری فعلی در شبکه‌ی بیت‌کوین [تاریخ انتشار این کتاب]، یک معدنچی برای یافتن نونس مناسب باید در هر ثانیه نزدیک به ۵۰ میلیارد-میلیارد دَرهم را امتحان کند.

در مثال ۱۰-۱۱ یک پیاده‌سازی بسیار ساده از الگوریتم اثبات-کار که با زبان پایتون نوشته شده، مشاهده می‌کنید.

مثال ۱۰-۱۱ یک پیاده‌سازی ساده از الگوریتم اثبات-کار

```
#!/usr/bin/env python
# example of proof-of-work algorithm

import hashlib
import time

max_nonce = 2 ** 32 # 4 billion

def proof_of_work(header, difficulty_bits):
    # calculate the difficulty target
    target = 2 ** (256 - difficulty_bits)
```

```
    for nonce in xrange(max_nonce):
        hash_result = hashlib.sha256(str(header)+str(nonce)).hexdigest()

        # check if this is a valid result, below the target
        if long(hash_result, 16) < target:
            print "Success with nonce %d" % nonce
            print "Hash is %s" % hash_result
            return (hash_result,nonce)

    print "Failed after %d (max_nonce) tries" % nonce
    return nonce


if __name__ == '__main__':
    nonce = 0
    hash_result = ''

    # difficulty from 0 to 31 bits
    for difficulty_bits in xrange(32):
        difficulty = 2 ** difficulty_bits
        print "Difficulty: %ld (%d bits)" % (difficulty, difficulty_bits)

        print "Starting search..."

        # checkpoint the current time
        start_time = time.time()

        # make a new block which includes the hash from the previous block
        # we fake a block of transactions - just a string
        new_block = 'test block with transactions' + hash_result

        # find a valid nonce for the new block
        (hash_result, nonce) = proof_of_work(new_block, difficulty_bits)

        # checkpoint how long it took to find a result
        end_time = time.time()

        elapsed_time = end_time - start_time
        print "Elapsed Time: %.4f seconds" % elapsed_time

        if elapsed_time > 0:
            # estimate the hashes per second
            hash_power = float(long(nonce)/elapsed_time)
            print "Hashing Power: %ld hashes per second" % hash_power
```

با این برنامه می‌توانید سطح دشواری دلخواه (تعداد بیت‌های پیش‌آیند که باید در ابتدای رشته‌ی دَرهم صفر باشند) را انتخاب کنید و ببینید کامپیوتر شما برای محاسبه‌ی پاسخ مناسب چه مقدار زمان صرف می‌کند. در مثال ۱۰-۱۲ اجرای این برنامه روی یک لپ‌تاپ معمولی نشان داده شده است.

مثال ۱۰-۱۲ اجرای الگوریتم اثبات-کار با دشواری‌های مختلف

```
# python proof-of-work-example.py*

Difficulty: 1 (0 bits)

[...]

Difficulty: 8 (3 bits)
Starting search...
Success with nonce 9
Hash is 1c1c105e65b47142f028a8f93ddf3dabb9260491bc64474738133ce5256cb3c1
Elapsed Time: 0.0004 seconds
Hashing Power: 25065 hashes per second
Difficulty: 16 (4 bits)
Starting search...
Success with nonce 25
Hash is 0f7becfd3bcd1a82e06663c97176add89e7cae0268de46f94e7e11bc3863e148
Elapsed Time: 0.0005 seconds
Hashing Power: 52507 hashes per second
Difficulty: 32 (5 bits)
Starting search...
Success with nonce 36
Hash is 029ae6e5004302a120630adcbb808452346ab1cf0b94c5189ba8bac1d47e7903
Elapsed Time: 0.0006 seconds
Hashing Power: 58164 hashes per second

[...]

Difficulty: 4194304 (22 bits)
Starting search...
Success with nonce 1759164
Hash is 0000008bb8f0e731f0496b8e530da984e85fb3cd2bd81882fe8ba3610b6cefc3
Elapsed Time: 13.3201 seconds
Hashing Power: 132068 hashes per second
Difficulty: 8388608 (23 bits)
Starting search...
Success with nonce 14214729
Hash is 000001408cf12dbd20fcba6372a223e098d58786c6ff93488a9f74f5df4df0a3
Elapsed Time: 110.1507 seconds
Hashing Power: 129048 hashes per second
Difficulty: 16777216 (24 bits)
Starting search...
Success with nonce 24586379
```

```
Hash is 0000002c3d6b370fccd699708d1b7cb4a94388595171366b944d68b2acce8b95
Elapsed Time: 195.2991 seconds
Hashing Power: 125890 hashes per second

[...]

Difficulty: 67108864 (26 bits)
Starting search...
Success with nonce 84561291
Hash is 0000001f0ea21e676b6dde5ad429b9d131a9f2b000802ab2f169cbca22b1e21a
Elapsed Time: 665.0949 seconds
Hashing Power: 127141 hashes per second
```

همان طور که می‌بینید، یک بیت افزایش سطح دشواری باعث دو برابر شدن زمان جستجو برای یافتن پاسخ (دَرهم مناسب) می‌شود. اگر کل فضای اعداد ۲۵۶-بیتی را در نظر بگیرید، هر بار که یک بیت بیشتر را مجبور به صفر بودن می‌کنید، در واقع فضای جستجو را نصف کرده‌اید. در مثال ۱۰-۱۲، برای یافتن نونسی که یک دَرهم با ۲۶ بیت پیش‌آیند صفر تولید کند، باید ۸۴ میلیون دَرهم مختلف محاسبه و امتحان شوند. حتی با سرعت بیش از ۱۲۰ هزار دَرهم بر ثانیه، یافتن پاسخ (روی این کامپیوتر) بیش از ۱۱ دقیقه زمان برده است.

در زمان انتشار این کتاب، گره‌های معدنچی باید بلاکی را جستجو کنند که دَرهم سرآیند آن کوچکتر از 0000000000000000000000000000000000000029ff380 باشد. همان طور که می‌بینید، تعداد زیادی صفر در ابتدای این «هدف» وجود دارد، بنابراین یافتن یک دَرهم که کوچکتر از آن باشد، بسیار دشوار است. برای یافتن دَرهم مناسب (و استخراج بلاک بعدی) به طور میانگین باید بیش از ۳ هزار میلیارد- میلیارد (۳۰۰۰ EH) دَرهم محاسبه و امتحان شود. این شاید کاری غیرممکن به نظر برسد، ولی خوشبختانه با قدرت پردازش فعلی شبکه‌ی بیت‌کوین (۵۰ میلیارد- میلیارد دَرهم بر ثانیه، ۵۰ EH/sec، در زمان انتشار کتاب) زمان یافتن یک بلاک به طور میانگین ۱۰ دقیقه خواهد بود. [در حقیقت، سطح دشواری در شبکه‌ی بیت‌کوین همیشه به طور خودکار به گونه‌ای تنظیم می‌شود که میانگین زمان استخراج یک بلاک جدید حدود ۱۰ دقیقه باشد.]

نمایش هدف

در مثال ۱۰-۳ دیدیم که در سرآیند بلاک ۲۷۷۳۱۶ فیلدی به نام «بیت‌های هدف» [یا فقط «بیت‌ها»] با مقدار 0x1903a30c وجود دارد. این فیلد هدف اثبات-کار را با فرمت ضریب/توان (coefficient/exponent) بیان می‌کند، که در آن دو رقم هگزادسیمال اول توان و شش رقم هگزادسیمال بعدی ضریب (پایه) را مشخص می‌کنند. به عبارت دیگر، در بلاک ۲۷۷۳۱۶ توان هدف 0x19 و ضریب آن 0x03a30c است.

فرمول محاسبه‌ی «هدف دشواری» از این مقدار چنین است:

```
target = coefficient * 2^(8 * (exponent - 3))
```

برای محاسبه‌ی بیت‌های [هدف] دشواری برای مقدار 0x1903a30c (بلاک ۲۷۷۳۱۶)، خواهیم داشت:

```
target = 0x03a30c * 2^(0x08 * (0x19 - 0x03))^
=> target = 0x03a30c * 2^(0x08 * 0x16)^
=> target = 0x03a30c * 2^0xB0^
```

که در دستگاه دهدهی (اعشاری) به صورت زیر در می‌آید:

```
target = 238,348 * 2^176^
=> target =
22,829,202,948,393,929,850,749,706,076,701,368,331,072,452,018,388,575,715,328
```

و اگر دوباره آن را به هگزادسیمال تبدیل کنیم، به هدف دشواری زیر می‌رسیم:

```
target = 0x00000000000000003A30C00000000000000000000000000000000000000000000000
```

به عبارت دیگر، بلاکی که بخواهد برای ارتفاع ۲۷۷۳۱۶ معتبر تلقی شود، باید سرآیندی کوچکتر از این هدف داشته باشد. در دستگاه باینری این عدد با بیش از ۶۰ صفر شروع می‌شود. با این سطح دشواری، یک گره معدنچی با قدرت پردازش هزار میلیارد دَرهم بر ثانیه (۱ TH/sec) به طور میانگین فقط در هر ۸۴۹۶ بلاک یا هر ۵۹ روز یک پاسخ (دَرهم مناسب) خواهد یافت.

بازتنظیم هدف برای تغییر سطح دشواری

همان طور که گفتیم، سطح دشواری عملیات جستجوی پاسخ الگوریتم اثبات-کار و همچنین زمان یافتن این پاسخ توسط «هدف» مشخص می‌شود. اما این ویژگی پرسش‌هایی را پیش می‌آورد: چرا دشواری شناور (قابل تنظیم) است؟ چه کسی مسئول این کار است؟ و چگونه آن را تنظیم می‌کند؟

بلاک‌های بیت‌کوین به طور میانگین هر ۱۰ دقیقه تولید می‌شوند. این آهنگ ضربان قلب بیت‌کوین است که سرعت انتشار بیت‌کوین‌های جدید و سرعت تسویه‌ی تراکُنش‌ها را مشخص می‌کند. [درست مثل ضربان قلب انسان] این آهنگ باید در کوتاه‌مدت و همچنین در بلندمدت (در طول چند دهه) ثابت بماند. اما سرعت پردازش کامپیوترها و تجهیزات سخت‌افزاری با آهنگی شتابان در حال افزایش است، و علاوه بر آن در طی دهه‌ی گذشته تعداد افراد و کامپیوترهایی که دست به استخراج بلاک‌های بیت‌کوین می‌زنند، نیز به طور پیوسته و با آهنگی خیره‌کننده بالا رفته است. برای آن که زمان تولید بلاک‌های جدید بیت‌کوین در همان محدوده‌ی ۱۰ دقیقه باقی بماند، لازم است دشواری استخراج متناسب با این تغییرات تنظیم شود. در واقع، هدف اثبات-کار یک پارامتر دینامیک است که به طور منظم بازتنظیم می‌شود تا معیار «هر ۱۰ دقیقه یک بلاک» را برآورده کند. به زبان ساده، *هدف دشواری به گونه‌ای تنظیم می‌شود که با توان پردازش موجود در سیستم بیت‌کوین در همان لحظه، بلاک‌های جدید در فواصل ۱۰-دقیقه‌ای تولید شوند.*

اما در یک شبکه‌ی کاملاً غیرمتمرکز این اتفاق چگونه رُخ می‌دهد؟ **بازتنظیم هدف** (retargeting) به طور خودکار در هر گره و به صورت مستقل انجام می‌شود. در هر ۲۰۱۶ بلاک، تمامی گره‌های شبکه هدف اثبات-کار را بازتنظیم می‌کنند. «معادله‌ی بازتنظیم هدف» زمان صرف‌شده برای یافتن آخرین ۲۰۱۶ بلاک را اندازه گرفته و آن را با زمان موردانتظار ۲۰۱۶۰ دقیقه (برای استخراج ۲۰۱۶ بلاک در فواصل ۱۰-دقیقه‌ای) مقایسه می‌کند. بعد از محاسبه‌ی نسبت زمان واقعی (x دقیقه) به زمان مطلوب (۲۰۱۶۰ دقیقه = ۲ هفته)، تنظیمات مورد نیاز در هدف (کم یا زیاد کردن سطح دشواری) اِعمال می‌شود. به بیان ساده: اگر میانگین سرعت یافتن بلاک‌های جدید در شبکه‌ی بیت‌کوین کمتر از ۱۰ دقیقه باشد، دشواری افزایش داده می‌شود (هدف کوچکتر می‌شود)؛ ولی اگر میانگین سرعت کشف بلاک‌های جدید بیشتر از زمان موردانتظار باشد، دشواری کاهش داده خواهد شد (هدف بزرگتر می‌شود). این معادله را می‌توان چنین خلاصه کرد:

```
New Target = Old Target * (Actual Time of Last 2016 Blocks / 20160 minutes)
```

کُدپیاده‌سازی این فرمول در مشتری هسته‌ی بیت‌کوین را در مثال ۱۰-۱۳ مشاهده می‌کنید. [پارامترهای Interval (۲۰۱۶ بلاکی) و TargetTimespan (۲ هفته = ۱,۲۰۹,۶۰۰ ثانیه) در ماژول chainparams.cpp تعریف شده‌اند.]

مثال ۱۰-۱۳ بازتنظیم هدف اثبات-کار: تابع CalculateNextWorkRequired( ) در ماژول pow.cpp

```
// Limit adjustment step
int64_t nActualTimespan = pindexLast->GetBlockTime() - nFirstBlockTime;
LogPrintf(" nActualTimespan = %d before bounds\n", nActualTimespan);

if (nActualTimespan < params.nPowTargetTimespan/4)
    nActualTimespan = params.nPowTargetTimespan/4;
if (nActualTimespan > params.nPowTargetTimespan*4)
    nActualTimespan = params.nPowTargetTimespan*4;

// Retarget
const arith_uint256 bnPowLimit = UintToArith256(params.powLimit);
arith_uint256 bnNew;
arith_uint256 bnOld;
bnNew.SetCompact(pindexLast->nBits);
bnOld = bnNew;
bnNew *= nActualTimespan;
bnNew /= params.nPowTargetTimespan;

if (bnNew > bnPowLimit)
    bnNew = bnPowLimit;
```

نکته

هر چند بازتنظیم هدف هر ۲۰۱۶ بلاک یک بار رُخ می‌دهد، به خاطر وجود خطای «تفاوت-یک» در ویرایش اولیه‌ی هسته‌ی بیت‌کوین، هنگام محاسبه‌ی این فرمول زمان استخراج ۲۰۱۵ بلاک در نظر گرفته می‌شود. در نتیجه هدف جدید همیشه با پنج‌صدم (۰٫۰۵) درصد خطا به سمت دشواری بیشتر تنظیم خواهد شد.

برای جلوگیری از تغییرات شدید و افراطی در هدف دشواری الگوریتم اثبات-کار، میزان بازتنظیم به کمتر از مرتبه‌ی ۴ در هر دوره محدود شده است. اگر مقدار بازتنظیم مورد نیاز بیشتر از مرتبه‌ی ۴ باشد، این میزان در همان حد مرتبه‌ی ۴ (و نه بیشتر) حفظ شده، و تغییرات بزرگتر از مرتبه‌ی ۴ به دوره‌ی بازتنظیم بعدی منتقل می‌شوند، چون این عدم‌تعادل به ۲۰۱۶ بلاک بعدی نیز منتقل خواهد شد. بنابراین، گاهی اوقات از میان برداشتن (صاف کردن) اختلاف شدید بین قدرت پردازش (دَرهم‌سازی) شبکه و هدف دشواری می‌تواند چندین دوره‌ی ۲۱۰۶-بلاکی طول بکشد.

توجه

دشواری استخراج یک بلاک بیت‌کوین تقریباً برابر است با ۱۰ دقیقه پردازش برای کل شبکه‌ی بیت‌کوین، بر اساس میانگین زمان استخراج ۲۰۱۶ بلاک قبلی، که این دشواری در هر ۲۰۱۶ بلاک مجدداً تنظیم شده و با کوچک یا بزرگ کردن «هدف» حاصل می‌شود.

توجه داشته باشید که هدف دشواری به کلی از تعداد تراکُنش‌ها و مبلغ آنها مستقل است. به عبارت دیگر، قدرت پردازش شبکه‌ی بیت‌کوین و *انرژی مصرفی آن برای ایمن کردن بیت‌کوین* هیچ رابطه‌ای با تعداد تراکُنش‌ها ندارد. بیت‌کوین می‌تواند گسترش یابد، مقبولیت بیشتری پیدا کند، و همچنان اَمن بماند، بدون این که توان پردازشی (دَرهم‌سازی) بیشتری نسبت به امروز بخواهد.

افزایش قدرت درهم‌سازی شبکه فقط ناشی از ورود معدنچیان جدید به این بازار و رقابت شدیدتر برای کسب جایزه‌ی استخراج بلاک‌ها است. تا زمانی که معدنچیان صادق (آنهایی که فقط به دنبال این جایزه هستند) قدرت درهم‌سازی کافی در اختیار داشته باشند، حمله‌های «تصاحب» شبکه و کنترل آن بی‌اثر خواهند بود و بیت‌کوین به اندازه‌ی کافی امن باقی خواهد ماند.

دشواری فرآیند استخراج بیت‌کوین رابطه‌ی تنگاتنگی با هزینه‌ی انرژی (برق مصرفی) و نرخ تسعیر بیت‌کوین (به ارزی که برای پرداخت بهای برق به کار می‌رود) دارد. سیستم‌های استخراج پیشرفته از کارایی بالایی برخوردار هستند و درهم‌سازی را با بالاترین توان ممکن انجام می‌دهند. بنابراین، مهم‌ترین عامل در بازار استخراج بیت‌کوین هزینه‌ی هر کیلووات-ساعت برق مصرفی است، چون سودآوری استخراج بیت‌کوین و در نتیجه توجیه اقتصادی ورود به این بازار (یا خروج از آن) را تعیین می‌کند.

استخراج موفق بلاک نامزد

همان طور که دیدیم، گره جینگ یک بلاک نامزد ساخته و آن را آماده‌ی استخراج کرده است. جینگ صدها دستگاه تخصصی استخراج بیت‌کوین دارد که هر کدام مجهز به صدها تراشه‌ی ASIC هستند، مدارهایی که انحصاراً برای اجرای الگوریتم SHA256 به صورت موازی و با سرعتی باورنکردنی (آزمایش دهها تریلیون [هزار میلیارد] نونس در هر ثانیه!) طراحی شده‌اند. این دستگاه‌های تخصصی از طریق ارتباط USB یا LAN به گره جینگ [که معمولاً روی یک کامپیوتر معمولی با سیستم‌عامل‌های آشنا مثل لینوکس، MacOS یا ویندوز اجرا می‌شود] متصل می‌شوند. در مرحله‌ی بعد، گره جینگ سرآیند این بلاک را به سخت‌افزار استخراج بیت‌کوین می‌فرستد و آنها فرآیند استخراج را آغاز می‌کنند.

تقریباً ۱۱ دقیقه بعد از شروع استخراج بلاک [نامزد] ۲۷۷۳۱۶، یکی از ماشین‌های استخراج مزرعه‌ی جینگ پاسخ معما را یافته و آن را به گره جینگ می‌فرستد. وقتی این دستگاه سرآیند بلاک ۲۷۷۳۱۶ را با نونس ۹۲۴,۵۹۱,۷۵۲ درهم‌سازی می‌کند، به درهم 0000000000000001b6b9a13b095e96db41c4a928b97ef2d944a9b31b2cc7bdc4 می‌رسد که از هدف 0000000000000003A30C0000000000000000000000000000000000000000 کوچکتر است.

گره جینگ بلافاصله این بلاک را به تمامی گره‌های همسایه (همتا) خود ارسال می‌کند. آنها این بلاک را دریافت کرده و بعد از اعتبارسنجی، آن را به گره‌های دیگر می‌فرستند. با این روش، بلاک جدید به تدریج در شبکه‌ی بیت‌کوین منتشر شده، و هر گرهی که این بلاک را دریافت می‌کند، آن را به بلاک‌چین محلی خود اضافه کرده و ارتفاع آن را به ۲۷۷۳۱۶ می‌رساند. گره‌هایی که این بلاک را دریافت کرده و بعد از اعتبارسنجی، از معتبر بودن آن مطمئن می‌شوند، تلاش برای استخراج بلاک ۲۷۷۳۱۶ را قطع کرده و بلافاصله استخراج بلاک بعدی را (با استفاده از بلاک استخراج‌شده توسط جینگ به عنوان بلاک «مادر») شروع می‌کنند. با این عمل (ساختن بلاک جدید بر اساس بلاک تازه استخراج‌شده توسط جینگ)، آن گره‌های معدنچی در واقع به بلاک جینگ «رأی اعتماد» داده، آن را با قدرت پردازش خود تأیید (پشت‌نویسی) می‌کنند، و زنجیره‌ی بلاک‌چین را توسعه می‌دهند.

در قسمت بعد نگاهی می‌اندازیم به فرآیند اعتبارسنجی بلاک‌های جدید در هر گره و انتخاب بلندترین زنجیره، و ایجاد اجماعی که پایه و اساس بلاک‌چین غیرمتمرکز را شکل می‌دهد.

اعتبارسنجی یک بلاک جدید

سومین گام در ساز و کار اجماع بیت‌کوین عبارت است از اعتبارسنجی مستقل هر بلاک جدید توسط تک تک گره‌های شبکه‌ی بیت‌کوین. با انتشار بلاک تازه استخراج‌شده در شبکه، هر گره قبل از بازنشر آن، چند آزمایش روی این بلاک انجام می‌دهد. بدین ترتیب فقط بلاک‌های معتبر می‌توانند روی شبکه‌ی بیت‌کوین منتشر شوند. اعتبارسنجی مستقل بلاک‌های جدید همچنین تضمین می‌کند که معدنچیان درستکار بتوانند بلاک‌های خود را وارد بلاک‌چین کنند و به جایزه‌ای که مستحق آن هستند، دست یابند. بلاک‌های ادعایی آن دسته از معدنچیانی که بخواهند کلک بزنند، نه تنها رد شده و هیچ جایزه‌ای به آنها تعلق نمی‌گیرد، بلکه به خاطر برقی که برای تولید بلاک‌های معیوب مصرف کرده‌اند، ضرر هم خواهند کرد.

وقتی یک گره بلاک جدیدی دریافت می‌کند، آن را با معیارهای مختلفی محک می‌زند؛ بلاک‌هایی که از این آزمایش‌ها سربلند بیرون نیایند، مردود خواهند شد. این معیارها، که در دو تابع مشترک هسته‌ی بیت‌کوین به نام‌های CheckBlock و CheckBlockHeader پیاده‌سازی شده‌اند، چنین هستند:

- ساختمان داده‌ی بلاک از نظر دستوری صحیح باشد؛
- دَرهم سرآیند بلاک از «هدف» کوچکتر باشد (اثبات-کار محقق شود)؛
- برچسب زمان بلاک (به خاطر خطای اختلاف زمان) بیش از دو ساعت از زمان فعلی جلوتر نباشد؛
- اندازه‌ی بلاک در محدوده‌ی قابل‌قبول باشد؛
- تراکُنش اول بلاک (و فقط همین تراکُنش اول) یک تراکُنش پایگاه‌سکه باشد؛
- تمام تراکُنش‌های موجود در این بلاک (بر اساس معیارهایی که در ابتدای همین فصل گفتیم) معتبر باشند.

اعتبارسنجی مستقل هر بلاک جدید توسط تک تک گره‌های شبکه‌ی بیت‌کوین تضمین می‌کند که معدنچیان نتوانند تقلب کنند. در قسمت‌های قبل دیدیم که چگونه معدنچیان خودشان تراکُنش تصاحب جایزه‌ی بلاک جدید (تراکُنش پایگاه‌سکه) و کارمزد تراکُنش‌های موجود در آن را می‌نویسند. چه چیزی مانع از آن می‌شود تا یک معدنچی به جای جایزه‌ی معمول استخراج بلاک جدید (۱۲٫۵ بیت‌کوین در حال حاضر) چند صد بیت‌کوین برای خود طلب نکند؟ علت فقط آن است که بلاک‌های جدید بر اساس قواعد یکسان در سرتاسر شبکه اعتبارسنجی می‌شوند، و یک تراکُنش پایگاه‌سکه‌ی نامعتبر (مثلاً، با مبلغ جایزه‌ی بیش از حد) باعث نامعتبر شدن کل بلاک و مردودشدن آن خواهد شد؛ چنین بلاکی هرگز نمی‌تواند وارد دفتر کل (بلاک‌چین) شود. معدنچیان باید بلاک‌های جدید را بر اساس قواعد مشترک در کل شبکه بسازند، و آنها را با یافتن یک پاسخ صحیح برای معمای اثبات-کار استخراج کنند. برای این کار، آنها باید مقدار زیادی برق مصرف کنند، و اگر تقلب کنند، همه‌ی این تلاش‌ها و انرژی مصرف‌شده هدر خواهد رفت. درست به همین دلیل است که اعتبارسنجی مستقل بلاک‌ها یک عنصر کلیدی در اجماع غیرمتمرکز محسوب می‌شود.

ساخت و انتخاب زنجیره‌ی بلاک

آخرین گام در ساز و کار اجماع غیرمتمرکز بیت‌کوین تجمیع بلاک‌ها به صورت زنجیره و انتخاب زنجیره‌ی دارای قویترین اثبات-کار است. همین که یک گره بلاک جدید را اعتبارسنجی کرد، سعی می‌کند با متصل کردن این بلاک به بلاک‌چین یک زنجیره بسازد. گره‌های بیت‌کوین سه «مجموعه بلاک» نگه می‌دارند: بلاک‌هایی که به بلاک‌چین اصلی متصل هستند (زنجیره‌ی اصلی)، بلاک‌های تشکیل‌دهنده‌ی شاخه‌هایی که از بلاک‌چین اصلی منشعب شده‌اند (زنجیره‌های ثانویه)، و سرانجام بلاک‌هایی که دارای یک والد شناخته‌شده در هیچ یک از زنجیره‌های شناخته‌شده نیستند (بلاک‌های یتیم). بلاک‌های نامعتبر به محض عدم انطباق با هر یک از معیارهای اعتبارسنجی مردود می‌شوند، بنابراین در هیچ زنجیره‌ای قرار نمی‌گیرند.

زنجیره‌ی اصلی (main chain) در هر زمان به زنجیره‌ای از بلاک‌های معتبر گفته می‌شود که بیشترین مقدار اثبات-کار در آن انباشته شده است. در اکثر مواقع این همان زنجیره‌ای است که بیشترین تعداد بلاک را در خود تجمیع کرده است، مگر این که دو زنجیره با طول یکسان وجود داشته باشند که مقدار اثبات-کار در یکی از آنها بیشتر باشد. زنجیره‌ی اصلی همچنین شاخه‌هایی خواهد داشت که بلاک‌های عضو آنها «خواهر» بلاک‌های عضو زنجیره‌ی اصلی هستند. این بلاک‌ها معتبر هستند ولی بخشی از زنجیره‌ی اصلی نیستند. این بلاک‌ها برای ارجاع در آینده نگه داشته می‌شوند، برای مواقعی که یکی از آن زنجیره‌ها طولانی‌تر از زنجیره‌ی اصلی فعلی شود. زنجیره‌های ثانویه نتیجه‌ی استخراج تقریباً همزمان دو بلاک با ارتفاع یکسان هستند؛ در قسمت بعد چگونگی ایجاد زنجیره‌های ثانویه را خواهید دید.

وقتی بلاک جدیدی وارد یک گره می‌شود، گره دریافت‌کننده تلاش می‌کند در بلاک‌چین موجود جایی برای آن پیدا کند. این گره به فیلد «دَرهم بلاک قبلی» بلاک واردشده (که ارجاعی به بلاک والد آن است) نگاه کرده و سپس در بلاک‌چین محلی خود به دنبال این [بلاک] والد جستجو می‌کند. اکثر مواقع، بلاک والد ارجاع‌شده در بلاک تازه‌وارد درست در «نوک» زنجیره‌ی اصلی قرار دارد، بدین معنا که بلاک جدید می‌تواند به انتهای زنجیره‌ی اصلی اضافه شده و آن را توسعه دهد (طولانی‌تر کند). برای مثال، بلاک جدید ۲۷۷۳۱۶ در فیلد «دَرهم بلاک قبلی» خود به بلاک ۲۷۷۳۱۵ به عنوان بلاک والد اشاره می‌کند. اکثر گره‌هایی که بلاک ۲۷۷۳۱۶ را دریافت می‌کنند، بلاک ۲۷۷۳۱۵ را در «نوک» زنجیره‌ی اصلی خود دارند، بنابراین بلاک جدید را به انتهای زنجیره‌ی اصلی لینک (متصل) کرده و این زنجیره را توسعه می‌دهند.

گاهی اوقات (همان طور که در مبحث «انشعاب بلاک‌چین» در ادامه‌ی همین فصل خواهید دید)، بلاک جدید زنجیره‌ای را توسعه می‌دهد که زنجیره‌ی اصلی نیست. در این حالت، گره دریافت‌کننده این بلاک را به زنجیره‌ی ثانویه‌ای که در اختیار دارد، متصل کرده و سپس مقدار اثبات-کار موجود در این زنجیره را با زنجیره‌ی اصلی مقایسه می‌کند. اگر مقدار اثبات-کار انباشته در زنجیره‌ی ثانویه بیشتر از زنجیره‌ی اصلی باشد، این گره خود را با زنجیره‌ی ثانویه همراستا (reconverge) می‌کند، یعنی آن را به عنوان زنجیره‌ی اصلی خود انتخاب کرده و زنجیره‌ی اصلی سابق را به یک زنجیره‌ی ثانویه تبدیل می‌کند. اگر این گره یک گره معدنچی باشد، در مسابقه‌ی دور بعد بلاکی را خواهد ساخت که این زنجیره‌ی جدید (و بزرگتر) را توسعه دهد.

اگر بلاک مادر (والد) بلاک جدیدی که وارد یک گره شده است، در هیچ یک از زنجیره‌های آن یافت نشود، به عنوان یک «بلاک یتیم» تلقی خواهد شد. بلاک‌های یتیم در یک مخزن خاص، موسوم به «مخزن بلاک‌های یتیم»، نگهداری می‌شوند تا زمانی که بلاک مادر آنها از راه برسد. همین که بلاک مادر دریافت شده و به یک زنجیره متصل (لینک) شد، آن بلاک یتیم از مخزن بلاک‌های یتیم خارج شده و به بلاک والد متصل (و جزیی از یک زنجیره) می‌شود. بلاک‌های یتیم معمولاً زمانی پدیدار می‌شوند که دو بلاک با فاصله‌ی زمانی کوتاه استخراج شده، ولی با ترتیب معکوس (فرزند قبل از مادر) دریافت شوند.

با انتخاب زنجیره‌ی معتبر بیشترین-کار-انباشته، تمامی گره‌ها به تدریج به اجماع در سطح شبکه دست می‌یابند. ناهمخوانی‌های موقتی بین زنجیره‌ها به مرور زمان و با اضافه شدن کار برطرف شده و یکی از زنجیره‌ها بیش از بقیه توسعه می‌یابد. گره‌های معدنچی با قدرت پردازش خود و انتخاب زنجیره‌ای که بلاک بعدی را در آن استخراج می‌کنند، در واقع به آن زنجیره «رأی اعتماد» می‌دهند. وقتی آنها یک بلاک جدید استخراج کرده و آن زنجیره را توسعه می‌دهند، خود این بلاک جدید «رأی اعتماد» آنها به آن زنجیره است.

در قسمت بعد خواهید دید که ناهمخوانی بین زنجیره‌های رقیب (انشعاب‌ها) چگونه با ساز و کار انتخاب مستقل زنجیره‌ی بیشترین-کار-انباشته برطرف می‌شود.

انشعاب‌های بلاک‌چین

از آنجا که بلاک‌چین یک ساختمان داده‌ی غیرمتمرکز است، نسخه‌های مختلف آن همیشه یکدست نیستند. بلاک‌ها در زمان‌های مختلف به گره‌های مختلف وارد شده و باعث می‌شوند تا گره‌ها نماهای متفاوتی از بلاک‌چین داشته باشند. برای رفع این مشکل، هر گره همیشه تلاش می‌کند تا زنجیره‌ی دارای بیشترین-اثبات-کار، که به نام *بلندترین زنجیره* یا زنجیره‌ی بیشترین-کار-انباشته (greatest-cumulative-work chain) نیز شناخته می‌شود، توسعه دهد. با جمع زدن کار ثبت‌شده در تک تک بلاک‌های یک زنجیره، هر گره می‌تواند مقدار کل کار صرف‌شده برای ایجاد آن زنجیره را محاسبه کند. مادامی که همه‌ی گره‌ها زنجیره‌ی بیشترین-کار-انباشته را انتخاب کنند، شبکه‌ی جهانی بیت‌کوین به تدریج به یک حالت یکدست (سازگار) همگرا می‌شود.

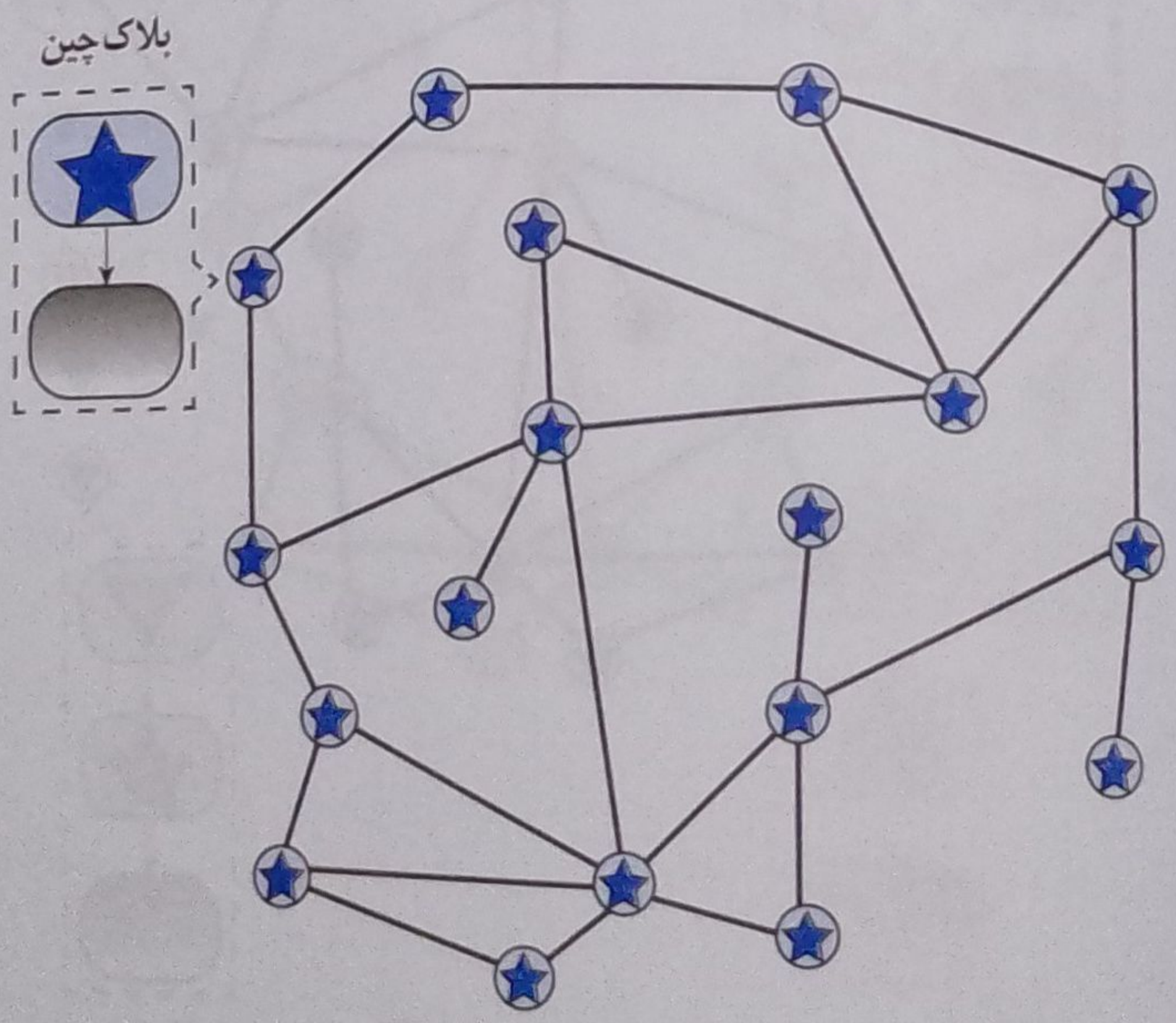
اما وقتی یک ناسازگاری موقتی بین نسخه‌های مختلف بلاک‌چین بروز کند، انشعاب (fork) رُخ می‌دهد؛ انشعاب‌ها معمولاً با همراستا شدن گره‌ها، در اثر اضافه شدن بلاک‌های بیشتر به یک انشعاب خاص، به تدریج برطرف می‌شوند.

بحث

انشعاب‌های بلاک‌چین که در این قسمت توضیح می‌دهیم، معمولاً به دلیل تأخیر انتشار در شبکه‌ی جهانی بروز می‌کنند. در ادامه‌ی همین فصل به انشعاب‌های عمدی هم خواهیم پرداخت.

در سلسله نمودارهایی که در این قسمت خواهید دید، روند ایجاد و گسترش (انتشار) یک انشعاب در کل شبکه نشان داده شده است (البته این نمودارها نمایی بسیار ساده از شبکه‌ی بیت‌کوین نشان می‌دهند). برای درک بهتر این نمودارها و روند تحول انشعاب، بلاک‌های متفاوتی که در شبکه پخش می‌شوند، با اَشکال هندسی مختلف (ستاره، مثلث، مثلث وارونه، لوزی)، و گره‌های شبکه با دایره تصویر شده‌اند.

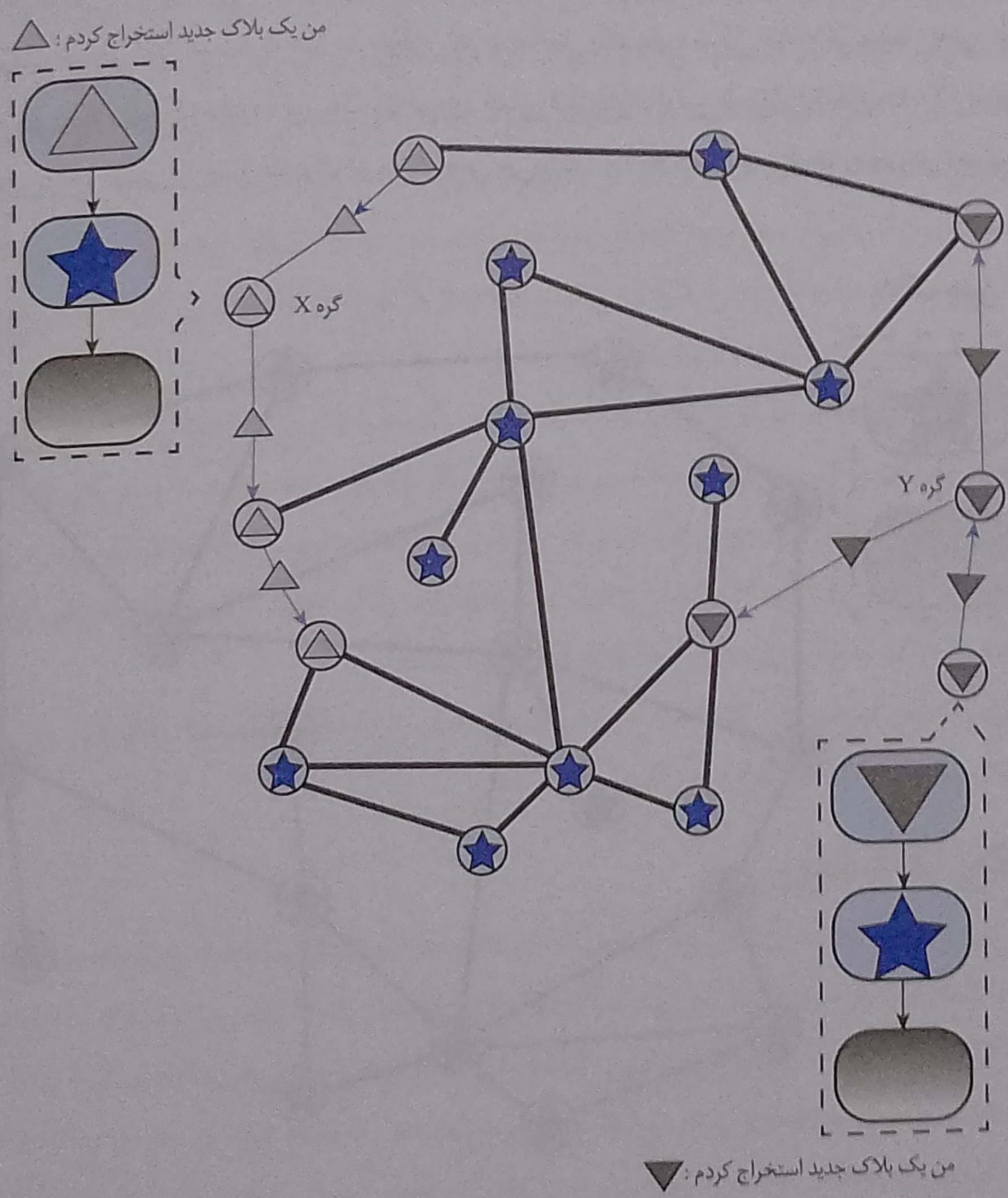
هر گره یک نسخه‌ی محلی (مخصوص به خود) از بلاک‌چین جهانی در اختیار دارد. با دریافت هر بلاک جدید از گره‌های همسایه، این گره با انتخاب زنجیره‌ی بیشترین-کار-انباشته، بلاک‌چین محلی خود را به‌روز می‌کند. شکل هندسی که در داخل هر دایره (گره) دیده می‌شود، بلاکی را نمایش می‌دهد که این گره تصور می‌کند نوک زنجیره‌ی اصلی (بلاک‌چین محلی) آن است؛ به عبارت دیگر، اگر در داخل یک دایره شکل ستاره می‌بینید، یعنی تا جایی که به این گره مربوط می‌شود، «بلاک ستاره» در نوک زنجیره‌ی اصلی قرار دارد. اولین نمودار (شکل ۱۰-۲) مربوط به زمانی است که همه‌ی گره‌ها نمایی واحد و مشترک از بلاک‌چین می‌بینند، و «بلاک ستاره» نوک زنجیره‌ی اصلی است.



شکل ۱۰-۲ قبل از انشعاب: تمام گره‌ها نمایی واحد از بلاک‌چین دارند.

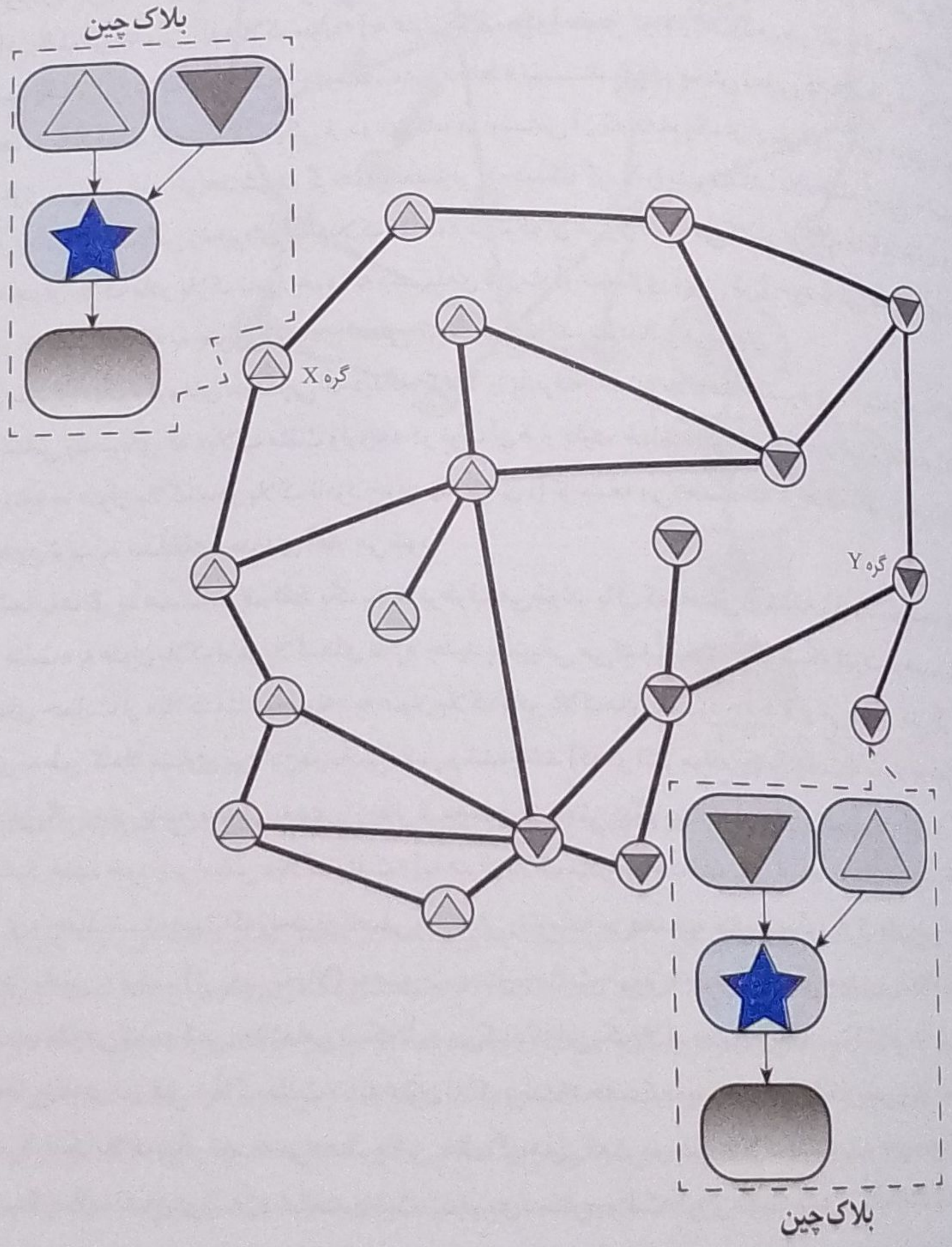
«انشعاب» زمانی رُخ می‌دهد که دو بلاک نامزد برای تشکیل بلندترین زنجیره رقابت کنند. در شرایط معمول، این حالت وقتی پیش می‌آید که دو گره معدنچی با فاصله‌ای کوتاه از یکدیگر معمای الگوریتم اثبات-کار را حل کنند. با یافتن پاسخ معما، هر دو گره بلافاصله بلاک «برنده» خود را روی شبکه منتشر می‌کنند. همان طور که قبلاً گفتیم، هر گرهی که یک بلاک [نامزد] معتبر دریافت می‌کند، بعد از اعتبارسنجی این بلاک، آن را به بلاک‌چین خود اضافه کرده و زنجیره‌ی اصلی را یک بلاک توسعه می‌دهد. اگر این گره کمی بعد بلاک نامزد دیگری با همان بلاک والد ببیند، بلاک نامزد دوم را به زنجیره‌ی ثانویه متصل می‌کند. به بیان دیگر، برخی گره‌ها یکی از این دو بلاک نامزد را اول «می‌بینند»، در حالی که دیگر گره‌ها بلاک نامزد دیگر را در نوک زنجیره‌ی اصلی خود خواهند دید، و در نتیجه دو ویرایش متفاوت از بلاک‌چین به وجود می‌آید.

در شکل ۱۰-۳، دو معدنچی (گره X و گره Y) می‌بینید که دو بلاک مختلف را تقریباً همزمان استخراج کرده‌اند. این بلاک‌ها هر دو فرزند «بلاک ستاره» هستند، و زنجیره را با ساخت یک بلاک جدید روی «بلاک ستاره» توسعه می‌دهند. برای تمایز بین این دو بلاک جدید، بلاک نامزد گره X را با مثلث، و بلاک نامزد گره Y را با مثلث وارونه نشان داده‌ایم.



شکل ۱۰-۳ وقوع انشعاب در بلاک‌چین: دو بلاک به طور همزمان استخراج می‌شوند.

وضعیت از این قرار است که: معدنچی گره X «بلاک مثلث» را به عنوان پاسخ معمای اثبات-کار که بلاک‌چین را روی «بلاک ستاره» توسعه می‌دهد، یافته است، و تقریباً در همان زمان معدنچی گره Y نیز «بلاک مثلث وارونه» را به عنوان پاسخ معمای اثبات-کار که آن هم بلاک‌چین را روی «بلاک ستاره» توسعه می‌دهد، می‌یابد. توجه کنید که این بلاک‌ها هر دو معتبر هستند، هر دو حاوی یک پاسخ معتبر برای معمای اثبات-کار هستند، و هر دو یک بلاک مادر واحد («بلاک ستاره») را توسعه می‌دهند. به احتمال زیاد اکثر تراکنش‌های هر دو بلاک نیز یکسان هستند، و شاید فقط ترتیب آنها کمی اختلاف داشته باشد. با انتشار این دو بلاک در شبکه، برخی گره‌ها ابتدا «بلاک مثلث»، و برخی دیگر ابتدا «بلاک مثلث وارونه» را دریافت می‌کنند. همان طور که در شکل ۱۰-۴ می‌بینید، بعد از مدتی شبکه به دو بخش متمایز با دو نمای متفاوت از بلاک‌چین تقسیم می‌شود: در یک بخش «بلاک مثلث» در نوک زنجیره‌ی اصلی (بلاک‌چین) قرار دارد، و در بخش دیگر «بلاک مثلث وارونه» نوک زنجیره‌ی اصلی است.



شکل ۱۰-۴ نمودار انشعاب در بلاک‌چین: دو بلاک در شبکه منتشر شده و آن را به دو بخش تقسیم می‌کنند.

در ایـــن نمودار، یک گره (که آن را «گره X» نامیده‌ایم) به طور تصادفی ابتدا «بلاک مثلث» را دریافت می‌کند و با این بلاک زنجیره‌ی «ستاره» را توسعه می‌دهد. گره X زنجیره‌ای را که «بلاک مثلث» در نوک آن قرار دارد، به عنوان زنجیره‌ی اصلی انتخاب می‌کند. کمی بعد، گره X «بلاک مثلث وارونه» را دریافت می‌کند؛ از آنجا که «بلاک مثلث وارونه» بعد از «بلاک مثلث» به این گره رسیده، از نظر گره X «بلاک مثلث وارونه» مسابقه را باخته است. با این حال، گره X «بلاک مثلث وارونه» را دور نمی‌اندازد [چون برنده‌ی نهایی هر دور از مسابقه بعد از استخراج ۱۰۰ بلاک به طور قطعی مشخص می‌شود]، بلکه آن را به «بلاک ستاره» به عنوان بلاک والد متصل کرده و یک زنجیره‌ی ثانویه ایجاد می‌کند. با آن که گره X از دیدگاه خودش زنجیره‌ی «برنده» را به درستی انتخاب کرده، ولی زنجیره‌ی «بازنده» را هم نگه می‌دارد تا اگر اوضاع تغییر کرد (جای برنده و بازنده‌ی این دور از مسابقه عوض شد)، بتواند خود را با سایر گره‌های شبکه همراستا کند.

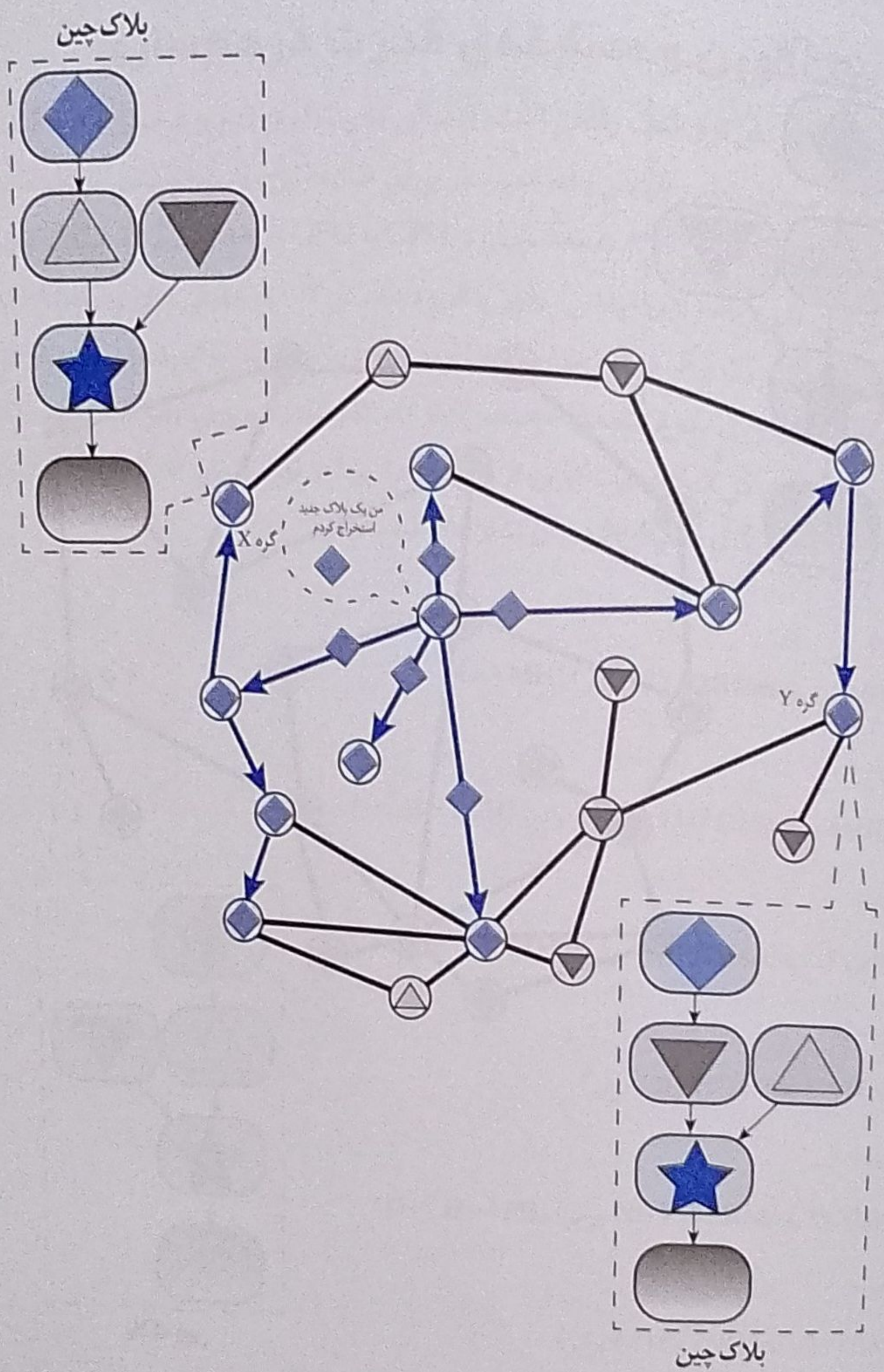
در ســـمت دیگر شـــبکه، گره Y بلاک‌چین را از دیدگاه خودش توسعه می‌دهد. این گره ابتدا «بلاک مثلث وارونه» را دریافت کرده و با این بلاک (به عنوان بلاک «برنده») زنجیره‌ی «ســـتاره» را توســـعه می‌دهد. وقتی کمی بعد این گره «بلاک مثلث» را دریافت می‌کند، آن را به «بلاک ستاره» [به عنوان بلاک مادر] متصل کرده و یک زنجیره‌ی ثانویه ایجاد خواهد کرد.

هیچ یک از این دو بخش شـــبکه «درســـت» یا «غلط» نیســـتند. هر دو بخش نمایی کاملاً معتبر از بلاک‌چین دارند. فقط در بلندمدت اســـت کـــه یکی از دو دیدگاه، بر اســـاس آن که کدام یک از ایـــن دو زنجیره موفق به جذب کار بیشـــتری شوند، برنده خواهد شـــد. گره‌های معدنچی که دیدگاه گره X را پذیرفته‌اند، بلافاصله شروع به استخراج بلاک‌های بعدی بر اســـاس زنجیره‌ای که «بلاک مثلث» در نوک آن قـــرار دارد، می‌کنند. این گره‌ها با انتخاب «بلاک مثلث» به عنوان بلاک مادر بلاک نامزد خود، به وســـیله‌ی قدرت دَرهم‌سازی [پردازش] خود به این بلاک رأی اعتماد می‌دهند و از زنجیره‌ای که به عنوان زنجیره‌ی اصلی انتخاب کرده‌اند، پشتیبانی می‌کنند.

از ســـوی دیگر، گره‌های معدنچی که دیدگاه گره Y را پذیرفته‌اند، نیز بلافاصله شـــروع به استخراج بلاک‌های بعدی بر اساس زنجیره‌ای که «بلاک مثلث وارونه» در نوک آن قرار دارد، خواهند کرد. این گره‌ها هم با انتخاب «بلاک مثلث وارونه» به عنوان بلاک مادر بلاک نامزد خود، زنجیره‌ای را توســـعه می‌دهنـــد که به اعتقاد آنها زنجیره‌ی اصلی است. و بدین ترتیب، مسابقه‌ی جدیدی آغاز می‌شود.

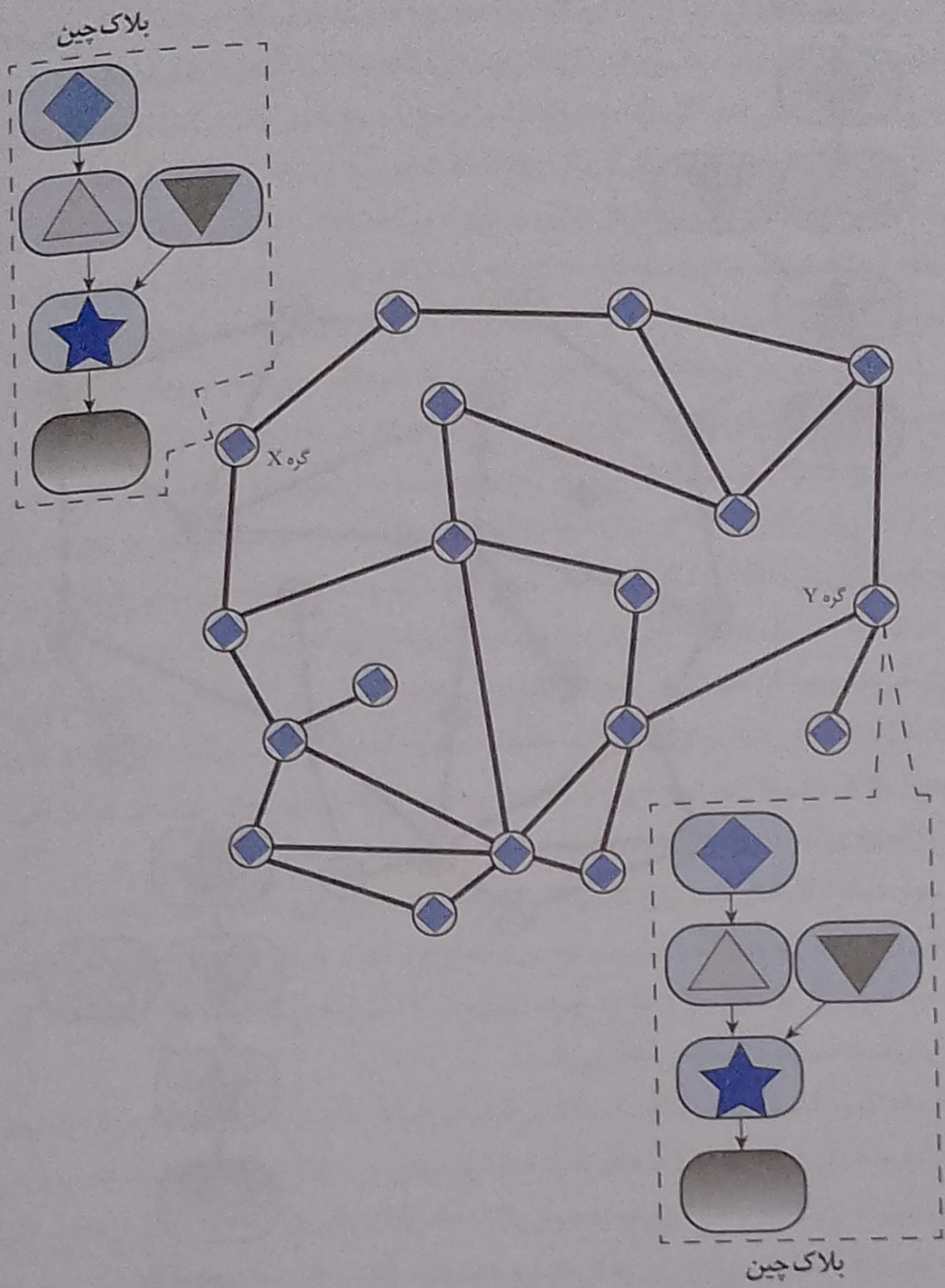
انشعاب‌ها تقریباً همیشه ظرف فقط یک بلاک برطرف می‌شوند. با آن که بخشی از شبکه با قدرت دَرهم‌سازی خود از «بلاک مثلث» به عنوان بلاک مادر بلاک‌های نامزد جدید پشتیبانی می‌کنند، بخش دیگر شبکه قدرت دَرهم‌سازی خود را پشـــتوانه‌ی حمایت از «بلاک مثلث وارونه» به عنوان بلاک مادر بلاک‌های نامـــزد جدید قرار می‌دهند. حتی اگر قدرت دَرهم‌سازی به طور کاملاً مساوی بین این دو بخش تقسیم شده باشد [که در اکثر مواقع چنین نیست]، بسیار محتمل است که معدنچیان یک بخش پاسخ معمای بعدی را زودتر از معدنچیان بخش دیگر پیدا کنند. برای مثال، فرض کنید در گروهی که بلاک نامزد جدید خود را بر اساس «بلاک مثلث» [به عنوان بلاک مادر] ساخته‌اند، یکی از معدنچیان موفق به یافتن یک «بلاک لوزی» جدید شـــده است که زنجیره‌ی اصلی آن بخش را توسعه می‌دهد؛ به عبارت دیگر، بلاک‌های رأس زنجیره‌ی اصلی در این بخش به ترتیب (از پایین به بالا) به صورت «ستاره- مثلث- لوزی» خواهد بود. این معدنچی بلافاصله بلاک جدید خود را منتشر می‌کند، و کمی بعد تمامی شبکه تأیید می‌کنند که این یک بلاک معتبر (برنده) است؛ شکل ۱۰-۵ را ببینید.

گره‌هایی که در دور قبل «بلاک مثلث» را به عنوان بلاک برنده انتخاب کرده بودند، هیچ مشکلی نمی‌بینند و به سادگی زنجیره‌ی خود را یک بلاک دیگر توسعه می‌دهند. با این حال، گره‌هایی که در دور قبل «بلاک مثلث وارونه» را به عنوان بلاک برنده انتخاب کرده‌اند، اکنون دو زنجیره خواهند داشت: زنجیره‌ی «ستاره- مثلث- لوزی» و زنجیره‌ی «ستاره- مثلث وارونه». اما زنجیره‌ی «ستاره- مثلث- لوزی» بلندتر است و (نسبت به زنجیره‌ی دیگر) کار بیشتری در آن انباشته شده است. در نتیجه، آن گره‌ها هم زنجیره‌ی «ســـتاره- مثلث- لوزی» را به زنجیره‌ی اصلی خود تبدیل کرده و زنجیره‌ی «ستاره- مثلث وارونه» را به



شکل ۱۰-۵ انشعاب در بلاک‌چین: بلاک جدید یکی از دو انشعاب را توسعه می‌دهد و باعث همراستایی شبکه می‌شود.

زنجیره‌ی ثانویه تنزل درجه می‌دهند؛ شکل ۱۰-۶ را ببینید. به این پدیده *همراستایی زنجیره* گفته می‌شود، چون گره‌ها مجبور هستند دیدگاه خود نسبت به بلاک‌چین را اصلاح کنند و استخراج در راستای بلندترین زنجیره را ادامه دهند. هر گره معدنچی که تا این لحظه مشغول توسعه‌ی زنجیره‌ی «ستاره- مثلث وارونه» بوده، کار خود را متوقف می‌کند چون بلاک نامزد آنها اکنون «یتیم» است و زنجیره‌ی مادر آن (زنجیره‌ی «ستاره- مثلث وارونه») دیگر بلندترین زنجیره نیست. تراکُنش‌های داخل «بلاک مثلث وارونه» نیز به مخزن تراکُنش برگردانده می‌شوند تا در بلاک‌های بعدی گنجانده شوند، چون بلاک آنها دیگر در زنجیره‌ی اصلی قرار ندارد. اکنون تمام شبکه روی بلاک‌چین «ستاره- مثلث- لوزی»، با «بلاک لوزی» در نوک زنجیره، همراستا شده است، و همه‌ی معدنچیان شروع به استخراج بلاک‌های جدید با استفاده از «بلاک لوزی» به عنوان بلاک مادر می‌کنند و زنجیره‌ی «ستاره- مثلث- لوزی» را توسعه می‌دهند.



شکل ۱۰-۶ انشعاب در بلاک‌چین: شبکه روی بلندترین زنجیره‌ی جدید به اجماع می‌رسد و همراستا می‌شود.

از لحاظ نظری این امکان وجود دارد که یک انشعاب تا دو بلاک هم توسعه یابد، به شرط آن که در دو استخراج متوالی، بلاک‌ها در هر دو بخش شبکه تقریباً به طور همزمان استخراج شوند. با این حال، احتمال بروز چنین پدیده‌ای بسیار اندک است. در واقع، انشعاب‌هایی که بعد از یک بلاک برطرف می‌شوند، تقریباً هر روز در شبکه رُخ می‌دهند، در حالی که انشعاب‌هایی که دو بلاک دوام می‌آورند، فقط هر چند هفته یا چند ماه یک بار ممکن است پیش آیند.

بازه‌ی زمانی ۱۰-دقیقه‌ای استخراج بلاک‌های جدید بیت‌کوین به گونه‌ای در نظر گرفته شده که بین تأیید (تسویه) سریع تراکُنش‌ها و احتمال بروز انشعاب تعادل ایجاد کند. اگر بلاک‌ها سریع‌تر استخراج شوند، تراکُنش‌ها نیز سریع‌تر تسویه می‌شوند، ولی انشعاب‌های بیشتری در بلاک‌چین رُخ خواهد داد؛ از سوی مقابل، زیاد کردن بازه‌ی استخراج بلاک‌های جدید می‌تواند آهنگ وقوع انشعاب در بلاک‌چین را کاهش دهد، ولی باعث کُند شدن تسویه‌ی تراکُنش‌ها خواهد شد.

استخراج بیت‌کوین و مسابقه‌ی قدرت دَرهم‌سازی

استخراج بیت‌کوین یک حوزه‌ی فوق‌العاده رقابتی است. از سالی که بیت‌کوین قدم به عرصه‌ی وجود گذاشت، قدرت دَرهم‌سازی هر سال به صورت نمایی افزایش یافته است. در برخی سال‌ها این رقابت انعکاسی از یک انقلاب فناوری بوده است، مانند سال‌های ۲۰۱۰ و ۲۰۱۱ که معدنچیان از CPU به GPU (واحد پردازش گرافیکی: کارت گرافیک) و FPGA (آرایه‌ی گیت قابل‌برنامه‌ریزی میدانی) تغییر پلتفرم دادند. در ۲۰۱۳، معرفی فناوری ASIC (مدار مجتمع برنامه-محور) به یک پرش بزرگ دیگر در قدرت استخراج بیت‌کوین انجامید: با این فناوری می‌توان کل الگوریتم دَرهم‌سازی SHA256 را روی یک تراشه‌ی تخصصی قرار داد. قدرت دَرهم‌سازی اولین تراشه‌های ASIC معادل مجموع قدرت دَرهم‌سازی کل شبکه‌ی بیت‌کوین در سال ۲۰۱۰ بود! در فهرست زیر آهنگ رشد قدرت دَرهم‌سازی شبکه‌ی بیت‌کوین در ده سال اول عملیاتی شدن آن نشان داده شده است:

۲۰۰۹ (۱۳۸۸)

از ۰٫۵ MH/sec به ۸ MH/sec (۱۶ برابر) [10^6 H = ۱ MH]

۲۰۱۰ (۱۳۸۹)

از ۸ MH/sec به ۱۱۶ GH/sec (۱۴۵۰۰ برابر) [10^9 H = ۱ GH]

۲۰۱۱ (۱۳۹۰)

از ۱۱۶ GH/sec به ۹ TH/sec (۷۸ برابر) [10^{12} H = ۱ TH]

۲۰۱۲ (۱۳۹۱)

از ۹ TH/sec به ۲۳ TH/sec (۲٫۵ برابر)

۲۰۱۳ (۱۳۹۲)

از ۲۳ TH/sec به ۱۰ PH/sec (۴۵۰ برابر) [10^{15} H = ۱ PH]

۲۰۱۴ (۱۳۹۳)

از ۱۰ PH/sec به ۳۰۰ PH/sec (۳۰ برابر)

۲۰۱۵ (۱۳۹۴)

از ۳۰۰ PH/sec به ۸۰۰ PH/sec (۲٫۶۶ برابر)

۲۰۱۶ (۱۳۹۵)

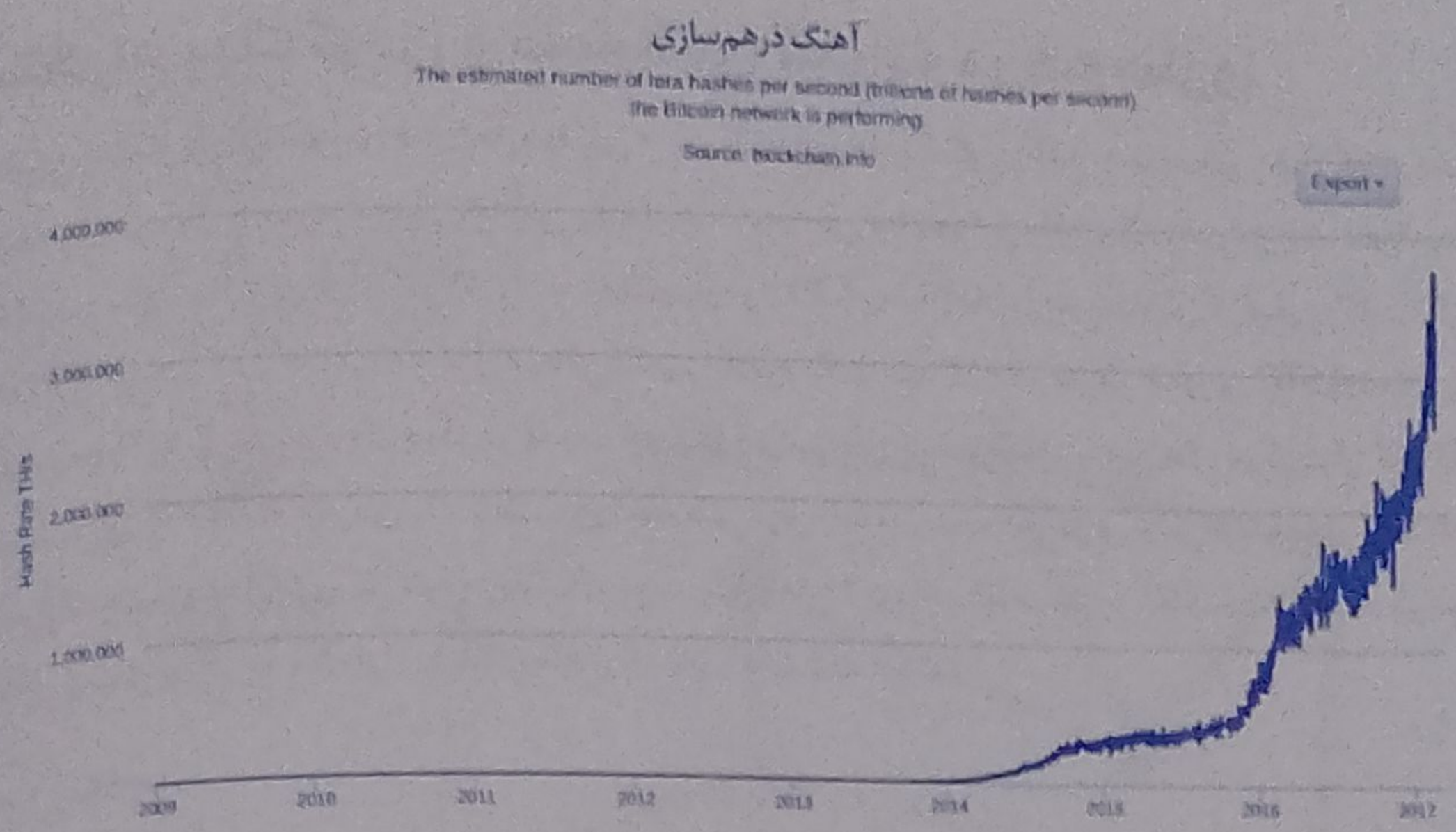
از ۸۰۰ PH/sec به ۲٫۵ EH/sec (۳٫۱۲ برابر) [10^{18} H = ۱ EH]

۲۰۱۷ (۱۳۹۶)

از ۲٫۵ EH/sec به ۱۷ EH/sec (۶٫۸ برابر)

۲۰۱۸ (۱۳۹۷)

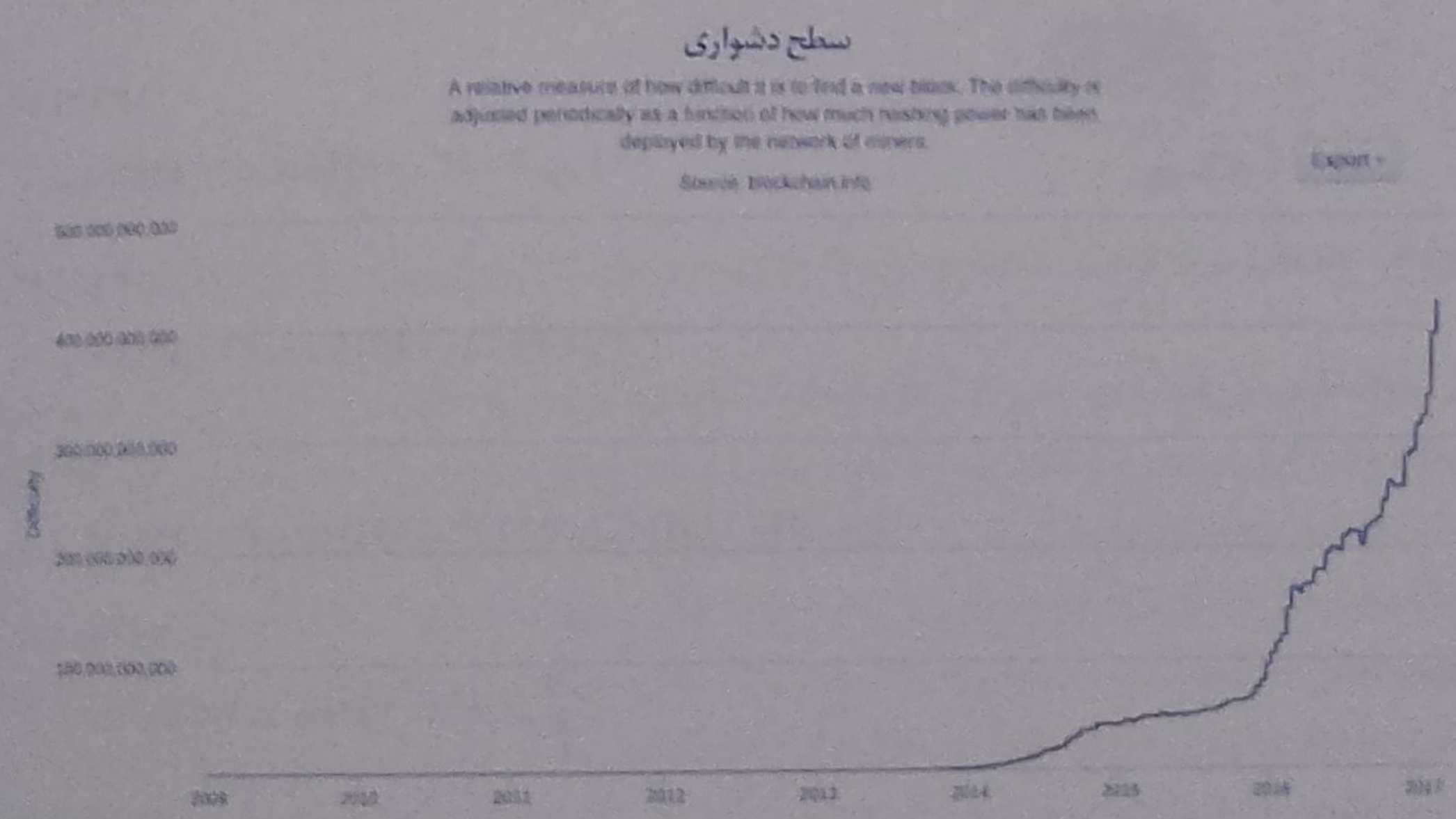
از ۱۷ EH/sec به ۴۸ EH/sec (۲٫۸ برابر)



شکل ۱۰-۷ قدرت دَرهم‌سازی کل شبکه‌ی بیت‌کوین، بر حسب TH/sec.

نمودار شکل ۱۰-۷ اوج‌گیری ناگهانی و شدید قدرت دَرهم‌سازی شبکه‌ی بیت‌کوین در سال‌های اخیر را نشان می‌دهد. همان طور که می‌توانید ببینید، رقابت بین معدنچیان و رشد قیمت بیت‌کوین منجر به جهش‌های نمایی چشمگیر در قدرت دَرهم‌سازی مورداستفاده در کل شبکه‌ی بیت‌کوین شده است.

با افزایش انفجاری قدرت دَرهم‌سازی به کار رفته در شبکه‌ی بیت‌کوین، دشواری معمای رمزنگاری آن نیز بالا رفته است تا زمان استخراج یک بلاک در همان محدوده‌ی ثابت ۱۰ دقیقه باقی بماند. نمودار شکل ۱۰-۸ میزان دشواری عملیات استخراج بیت‌کوین را به صورت نسبت «سطح دشواری فعلی» به «کمترین سطح دشواری» (دشواری استخراج اولین بلاک بیت‌کوین) نشان می‌دهد.



شکل ۱۰-۸ شاخص دشواری استخراج بیت‌کوین.

در سال‌های اخیر چگالی تراشه‌های ASIC رو به افزایش بوده است و امروزه در تولید آنها از پیشرفته‌ترین فناوری‌های ساخت تراشه، یعنی فناوری ۱۶-نانومتری، استفاده می‌شود و (به خاطر رقابت شدید و نیاز به قدرت پردازش بیشتر برای حفظ سودآوری عملیات استخراج) سازندگان مهم در حال طراحی تراشه‌های ASIC با فناوری ۱۴-نانومتری هستند. شاید (به دلیل سد محکمی به نام «قانون مور» که می‌گوید قدرت پردازش کامپیوتری هر ۱۸ ماه دو برابر می‌شود) ساخت تراشه‌های بسیار چگال‌تر به‌زودی ممکن نباشد، اما قدرت پردازش کل شبکه‌ی بیت‌کوین همچنان با آهنگی نمایی بالا خواهد رفت چون هر روز هزاران ماشین استخراج ASIC جدید به شبکه‌ی بیت‌کوین اضافه می‌شوند. امروزه دیگر صحبت از آن نیست که با یک تراشه چقدر بیت‌کوین (یا بهتر بگوییم: ساتوشی) می‌توان استخراج کرد، بلکه دغدغه‌ی امروز تعداد دستگاه‌هایی است که می‌توان در یک ساختمان گرد آورد، برای آنها برق کافی فراهم کرد، و گرمای تولیدشده در حرارت آنها را دفع کرد تا از شدت حرارت ذوب نشوند!

راه‌حل نونس‌های اضافی

از سال ۲۰۱۲، استخراج بیت‌کوین به سمتی رفت تا یکی از محدودیت‌های بنیادی در ساختار سرآیند بلاک را برطرف کند. در روزهای اولیه‌ی بیت‌کوین، یک معدنچی می‌توانست با تکرار از طریق تغییر دادن مقدار نونس (رشته‌ی تصادفی) و رسیدن به دَرهم کوچکتر از هدف، یک بلاک را پیدا (استخراج) کند. نونس یک عدد ۳۲-بیتی (۴-بایتی) است که می‌تواند کمی بیش از ۴ میلیارد مقدار متفاوت تولید کند، ولی با افزایش سطح دشواری عملیات استخراج، معدنچیان اغلب تمامی این ۴ میلیارد مقدار را امتحان می‌کردند، بدون این که بتوانند هیچ بلاکی پیدا (استخراج) کنند. با این حال، این مشکل به سادگی و با به‌روز کردن برچسب زمان بلاک (برای شمارش زمان سپری‌شده) حل شد. از آنجا که برچسب زمان بخشی از سرآیند بلاک است، این تغییر به معدنچیان اجازه می‌داد تا (با به‌روز شدن برچسب زمان بلاک) ۴ میلیارد مقدار متفاوت جدید برای نونس داشته باشند و بتوانند کار را با این مقادیر جدید ادامه دهند. اما همین که قدرت پردازش (دَرهم‌سازی) سخت‌افزارهای استخراج بیت‌کوین از ۴ GH/sec فراتر رفت، حتی این رویکرد نیز کارایی خود را از دست داد، چون دستگاه‌های استخراج جدید ۴ میلیارد نونس را در کمتر از یک ثانیه پردازش می‌کردند. با ورود دستگاه‌های استخراج ASIC که قدرت پردازش (دَرهم‌سازی) آنها در محدوده‌ی TH/sec (هزار میلیارد دَرهم در ثانیه!) است، کار حتی از قبل هم سخت‌تر هم شد، چون نرم‌افزار استخراج به بازه‌ی بزرگتری از مقادیر نونس احتیاج داشت تا بتواند یک بلاک را پیدا (استخراج) کند. یک روش دیگر این بود که برچسب زمان بلاک را به صورت دستی جلو ببرند تا بتوانند نونس‌های جدیدی تولید کنند، ولی این تمهید یک محدودیت جدی داشت: جلو بردن بیش از حد برچسب زمان بلاک نامزد باعث نامعتبر شدن آن می‌شد.

آشکار بود که معدنچیان بیت‌کوین به منبع جدیدی از «تغییر» در سرآیند بلاک نیاز داشتند. راه‌حل این مشکل در استفاده از تراکُنش پایگاه‌سکه به عنوان منبع نونس‌های اضافی نهفته بود. از آنجا که اسکریپت پایگاه‌سکه می‌تواند حاوی ۲ تا ۱۰۰ بایت داده باشد، معدنچیان شروع کردند به استفاده از این فضا به عنوان منبع نونس‌های اضافی، که به آنها اجازه می‌داد بازه‌های بسیار بزرگتری از مقادیر سرآیند بلاک را برای یافتن بلاک‌های معتبر جستجو کنند. از آنجا که تراکُنش پایگاه‌سکه در درخت مرکل گنجانده می‌شود، هر تغییری در اسکریپت پایگاه‌سکه باعث تغییر ریشه‌ی مرکل خواهد شد. با داشتن هشت بایت نونس اضافی، علاوه بر ۴ بایت نونس «استاندارد»، معدنچی می‌تواند در هر *ثانیه* (و بدون نیاز به دستکاری در برچسب زمان بلاک) بازه‌ای متشکل از 2^{96} (تقریباً 8×10^{28}) مقدار محتمل را جستجو کند. اگر (در آینده) این تعداد نونس هم برای پیدا کردن (استخراج) یک بلاک کافی نباشد، آنگاه معدنچی می‌تواند برچسب زمان بلاک را دستکاری کند. و این فقط هشت بایت از فضای داده‌ی اسکریپت پایگاه‌سکه است: به نظر می‌رسد این فضا تا مدت‌های طولانی برای تولید نونس‌های اضافی کافی باشد.

استخراج ائتلافی

در دنیای پُررقابت استخراج بیت‌کوین، معدنچیانی که به تنهایی کار می‌کنند (موسوم به معدنچیان تکرو)، شانس چندانی ندارند. احتمال این که آنها بتوانند یک بلاک جدید استخراج کنند (و حتی با هزینه‌ی برق مصرفی خود سر به سر کنند)، چنان پایین است که بیشتر به قمار یا لاتاری می‌ماند. حتی جدیدترین دستگاه‌های استخراج بیت‌کوین مجهز به سریع‌ترین تراشه‌های ASIC نمی‌توانند با مزرعه‌های بزرگ متشکل از دهها و صدها هزار ماشین استخراج بیت‌کوین که در سوله‌های مجاور نیروگاه‌های برق بر پا شده‌اند، رقابت کنند. امروزه معدنچیان شانس خود را با عضویت در ائتلاف (گروه) استخراج (mining pool) که قدرت پردازش و دَرهم‌سازی صدها هزار معدنچی مستقل را یکجا گرد آورده و جایزه‌ی استخراج بلاک‌های جدید را به تناسب بین آنها تقسیم می‌کنند، افزایش می‌دهند. با عضویت در یک ائتلاف استخراج، معدنچی سهم کوچکی (نسبت به کل جایزه‌ی استخراج یک بلاک) به دست می‌آورد، ولی این جایزه مستمر (به صورت روزانه) است، و دیگر شانس و اقبال چندان در آن دخیل نیست.

اجازه دهید این موضوع را با یک مثال خاص بیشتر تشریح کنیم. اوایل سال ۲۰۱۹ است، و یک دستگاه استخراج بیت‌کوین با قدرت پردازش ۱۴ هزار میلیارد دَرهم در ثانیه، ۱۴ TH/sec، چیزی حدود ۵۰۰ دلار قیمت دارد. توان مصرفی این دستگاه ۱۳۷۵ وات (۱٫۳۷۵ کیلووات، kW) است، که اگر یک شبانه‌روز به طور کامل کار کند، ۳۳ کیلووات-ساعت (kWH) انرژی الکتریکی مصرف خواهد کرد. با فرض کمترین قیمت انرژی الکتریکی، هزینه‌ی برق مصرفی این دستگاه در هر شبانه‌روز ۳ تا ۴ دلار خواهد بود. با سطح دشواری یافتن بلاک‌های جدید در این زمان (نزدیک به ۵۰ EH)، یک معدنچی تکرو شاید بتواند هر ۸۰ سال یک بلاک استخراج کند. جایزه‌ی استخراج یک بلاک در این زمان ۱۲٫۵ بیت‌کوین است، که با قیمت حدود ۷۵۰۰ دلاری بیت‌کوین [در همان دوره] به ۸۷۵۰۰ دلار بالغ خواهد شد. اما این معدنچی (بدون احتساب هزینه‌ی خرید، تعمیر و نگهداری دستگاه استخراج) باید بیش از ۱۰۰ هزار دلار برای برق مصرفی خود هزینه کند. اگر معدنچی ما خوش‌شانس باشد و بتواند در این مدت (به جای یک بلاک) ۲ بلاک استخراج کند، شاید بتواند به سودآوری برسد. ولی توجه کنید که دشواری استخراج بلاک‌های جدید (مقدار قدرت دَرهم‌سازی مورد نیاز برای استخراج یک بلاک در کل شبکه‌ی بیت‌کوین) هر سال چندین برابر افزایش می‌یابد، به طوری که دستگاه‌های استخراج بیت‌کوین تقریباً هر دو تا سه سال یک بار منسوخ و از رده خارج می‌شوند. جایزه‌ی استخراج بلاک جدید نیز هر چهار سال یک بار نصف می‌شود و در یک دوره‌ی ۸۰-ساله به چند صد ساتوشی تقلیل می‌یابد، که این عامل را هم در محاسبات بالا در نظر نگرفته‌ایم. اگر این معدنچی (به جای تکروی) عضو یک گروه استخراج شود، به جای ۸۷۵۰۰ دلار در هر ۸۰ سال، می‌تواند در هفته ۷۰ تا ۸۰ دلار درآمد مستمر و مطمئن داشته باشد که هزینه‌ی برق مصرفی و استهلاک دستگاه [با فرض ۳ سال عمر مفید] را به خوبی پوشش می‌دهد و سودآور هم هست (البته خطر این سرمایه‌گذاری همچنان بالا است). همان طور که می‌بینید، استخراج بیت‌کوین از نظر مالی فقط با انرژی بسیار ارزان (کمتر از ۵ تا ۷ سنت به ازای هر کیلووات-ساعت) و در مقیاس بسیار بزرگ (دهها و صدها هزار دستگاه استخراج) مقرون به صرفه است.

گروه‌های استخراج به کمک *پروتکل‌های استخراج-ائتلافی* هزاران دستگاه استخراج را به یکدیگر متصل و هماهنگ می‌کنند. معدنچیان مستقل، بعد از عضویت در یک ائتلاف و دریافت شناسه‌ی کاربری، دستگاه‌های استخراج خود را برای اتصال به این گروه پیکربندی می‌کنند.

این دستگاه‌ها در تمام مدت فرآیند استخراج به سرویس‌دهنده‌ی ائتلاف (pool server) که مسئول هماهنگ‌سازی دستگاه‌های مختلف است، متصل می‌مانند. از آنجا که تمامی دستگاه‌های عضو یک ائتلاف در استخراج بلاک‌های جدید شریک هستند، جایزه‌ی استخراج هم به تناسب بین آنها تقسیم می‌شود. این جایزه به آدرس بیت‌کوین ائتلاف پرداخت خواهد شد، ولی

سرویس‌دهنده‌ی ائتلاف در فواصل منظم (بعد از رسیدن مقدار هر سهم به یک آستانه‌ی معین) جایزه‌ی معدنچیان را به آدرس بیت‌کوین آنها پرداخت می‌کند. سرویس‌دهنده‌ی ائتلاف معمولاً درصدی از این جایزه را به عنوان کارمزد فراهم کردن سرویس استخراج-ائتلافی برای خود برمی‌دارد.

اعضای ائتلاف استخراج، کار جستجوی حل معمای رمزنگاری را بین خود تقسیم می‌کنند و هر کدام به نسبت کاری که انجام می‌دهند، «سهم» می‌گیرند. گروه‌های استخراج برای کسب یک سهم هدف دشواری بزرگتری (آسان‌تری) [معمولاً ۱۰۰۰ بار آسان‌تر از هدف دشواری شبکه‌ی بیت‌کوین] تعیین می‌کنند. وقتی یکی از اعضای ائتلاف موفق به استخراج یک بلاک می‌شود، جایزه‌ی آن بلاک به گروه تعلق گرفته و به نسبت تعداد سهم‌ها بین اعضا تقسیم می‌شود.

عضویت در گروه‌های استخراج معمولاً آزاد است و هیچ محدودیتی از نظر تعداد یا نوع دستگاه‌هایی که یک معدنچی می‌تواند وارد ائتلاف کند، وجود ندارد. به عبارت دیگر، یک ائتلاف می‌تواند عضوی با فقط یک دستگاه استخراج ساده و معمولی داشته باشد، یا عضوی که صدها دستگاه پیشرفته وارد ائتلاف کرده است. برخی از اعضای یک ائتلاف فقط چند ده کیلووات ساعت در روز برق می‌کنند، در حالی که مصرف اعضای دیگر ممکن است به اندازه‌ی یک شهرک کوچک (چندین مگاوات) باشد. یک ائتلاف استخراج چگونه میزان مشارکت هر عضو را می‌سنجد، به طوری که بتواند جوایز را منصفانه تقسیم کرده و جلوی تقلب را هم بگیرد؟ راه‌حل این مشکل در استفاده از الگوریتم اثبات-کار بیت‌کوین برای اندازه‌گیری میزان مشارکت هر عضو گروه نهفته است، البته با پایین آوردن سطح دشواری به گونه‌ای که حتی کوچکترین و ضعیف‌ترین معدنچیان هم بتوانند سهمی منظم به دست بیاورند و تشویق به عضویت و ماندن در ائتلاف شوند. با پایین آوردن سطح دشواری برای کسب سهمیه، ائتلاف استخراج مقدار کار انجام‌شده توسط هر عضو (معدنچی) را اندازه می‌گیرد. هر بار که یک معدنچی بلاک جدیدی استخراج می‌کند، ثابت می‌شود که دَرهم‌سازی مورد نیاز برای یافتن آن بلاک را انجام داده است. از آن مهم‌تر، این کار سهم مشخصی در تلاش کلی گروه برای یافتن هدف اصلی (پیدا کردن دَرهمی که از هدف تعیین‌شده توسط شبکه‌ی بیت‌کوین کوچکتر است) دارد که از نظر آماری قابل اندازه‌گیری است: هزاران معدنچی که در تلاش دستجمعی برای یافتن دَرهم‌های کوچک هستند، سرانجام یک دَرهم پیدا خواهند کرد که هدف شبکه‌ی بیت‌کوین را برآورده کند.

شاید جستجو برای هدفی بزرگتر (آسان‌تر) از هدف تعیین‌شده توسط شبکه‌ی بیت‌کوین و رسیدن به نتیجه‌ی مطلوب برای شما کمی عجیب و دور از ذهن باشد، پس اجازه دهید با همان مثال بازی تاس آن را بیشتر تشریح کنیم. اگر در یک دور از بازی هدف (دشواری) آوردن تاس‌هایی با مجموع کمتر از ۴ باشد، آنگاه گروه می‌تواند هدف داخلی خود را آسان‌تر انتخاب کند و (مثلاً) تعداد دفعاتی را که بازیکنان گروه تاس‌هایی با مجموع کمتر از ۸ می‌آورند، بشمارد. وقتی یک بازیکن کمتر از ۸ (هدف مورد نیاز برای کسب سهمیه در گروه) می‌آورد، سهمی از جایزه‌ی (احتمالی) برای وی ثبت می‌شود، ولی گروه هنوز بازی را نبرده است چون هدف بازی (کمتر از ۴) محقق نشده است. از آنجا که هدف (دشواری) داخلی گروه خیلی آسان‌تر از هدف بازی اصلی است، بازیکنان بیشتر و منظم‌تر سهمیه می‌گیرند، حتی وقتی به هدف بازی (که بسیار دشوارتر است) دست نیافته باشند. اما گروه از تعداد زیادی افراد مستقل تشکیل شده و هر از چند گاهی یکی از آنان تاسی با مجموع کمتر از ۴ می‌آورد و گروه برنده می‌شود. اکنون می‌توان جایزه‌ی بازی را به تناسب سهم هر نفر بین اعضای گروه تقسیم کرد. حتی اگر هدف «کمتر از ۸» برنده نباشد، استفاده از این هدف برای تعیین میزان مشارکت افراد در بازی کل گروه کافی است؛ و گاهی هم یک نفر هدف «کمتر از ۴» را می‌زند.

به طریق مشابه، یک ائتلاف استخراج بیت‌کوین هم هدفی بزرگتر (آسان‌تر) از هدف دشواری شبکه‌ی بیت‌کوین تعیین می‌کند تا مطمئن شود همه‌ی اعضای ائتلاف (قوی یا ضعیف) می‌توانند دَرهم‌های کوچکتر از این هدف را محاسبه کنند و سهمی از جایزه‌ی گروه به دست آورند. روزی چند بار هم شانس با ائتلاف همراهی می‌کند و یکی از اعضای ائتلاف دَرهمی کوچکتر از هدف دشواری شبکه‌ی بیت‌کوین پیدا کرده و باعث می‌شود تا کل ائتلاف برنده شود.

ائتلاف مدیریت‌شده

اکثر گروه‌های استخراج بیت‌کوین «مدیریت‌شده» هستند، یعنی یک فرد یا شرکت کنترل سرویس‌دهنده‌ی ائتلاف را در اختیار دارد. مالک سرویس‌دهنده‌ی ائتلاف که به آن راهبر ائتلاف (pool operator) گفته می‌شود، معمولاً درصدی از جایزه‌ی (احتمالی) را به عنوان کارمزد [فراهم کردن سرویس استخراج- ائتلافی] برای خود برمی‌دارد.

سرویس‌دهنده‌ی ائتلاف علاوه بر اجرای نرم‌افزارهای تخصصی به یک پروتکل استخراج- ائتلافی (pool-mining protocol) نیز برای هماهنگ کردن فعالیت اعضای گروه (معدنچیان مستقل) نیاز دارد. سرویس‌دهنده‌ی ائتلاف همچنین به یک یا چند گره کامل بیت‌کوین متصل است و دسترسی مستقیم به نسخه‌ی کاملی از پایگاه داده‌ی بلاک‌چین دارد. این دسترسی به سرویس‌دهنده‌ی ائتلاف اجازه می‌دهد تا بلاک‌ها و تراکنش‌ها را به نیابت از معدنچیان گروه اعتبارسنجی کند و بار سنگین راه‌اندازی و نگهداری یک گره کامل را از دوش آنان بردارد. این وضعیت برای معدنچیان عضو ائتلاف یک مزیت عمده محسوب می‌شود، چون راه‌اندازی گره کامل به یک کامپیوتر قوی با حداقل ۲۰۰ تا ۵۰۰ گیگابایت فضای دیسک و حداقل ۲ تا ۶ گیگابایت حافظه‌ی اصلی (RAM) نیاز دارد. علاوه بر آن، نرم‌افزارهای بیت‌کوین که در یک گره کامل اجرا می‌شوند، باید به طور منظم نظارت، نگهداری و به‌روز-رسانی شوند؛ هر خاموشی یا اختلالی که در اثر نگهداری نامناسب یا کمبود منابع سخت‌افزاری اتفاق بیفتد، به سودآوری کل ائتلاف لطمه خواهد زد. برای بسیاری از معدنچیان، توانایی استخراج بیت‌کوین بدون نیاز به راه‌اندازی و نگهداری یک گره کامل مهم‌ترین علت (انگیزه‌ی) پیوستن به یک ائتلاف استخراج مدیریت‌شده است.

معدنچیانی که می‌خواهند به یک سرویس‌دهنده‌ی ائتلاف متصل شوند، باید از یک پروتکل استخراج- ائتلافی مانند استراتوم (STM) یا **GetBlockTemplate** (GBT) استفاده کنند. استانداردهای قدیمی مانند GetWork (GWK) از اواخر ۲۰۱۲ به این طرف اکثراً منسوخ شده‌اند، چون قادر نیستند به سادگی از استخراج با سرعت‌های بالای ۴ GH/sec پشتیبانی کنند. هر دو پروتکل STM و GBT یک الگوی بلاک (block template) ایجاد می‌کنند که حاوی قالبی خالی از سرآیند بلاک نامزد است. سرویس‌دهنده‌ی ائتلاف با تجمیع تراکنش‌ها یک بلاک نامزد می‌سازد، تراکنش پایگاه‌سکه را (همراه با فضای نونس اضافی) به آن اضافه می‌کند، سپس ریشه‌ی مرکل بلاک را محاسبه کرده، و این بلاک را به دَرهم [سرآیند] بلاک قبلی متصل می‌کند. سپس این سرآیند [بلاک نامزد] به تمامی معدنچیان عضو گروه فرستاده می‌شود تا از آن به عنوان یک الگو استفاده کنند. هر معدنچی از این الگوی بلاک که هدف دشواری آن بزرگتر (آسان‌تر) از هدف دشواری شبکه‌ی بیت‌کوین است، برای استخراج بیت‌کوین استفاده می‌کند، و وقتی یک دَرهم کوچکتر از هدف دشواری تعیین‌شده در سرآیند این بلاک یافت، آن را به سرویس‌دهنده‌ی ائتلاف برمی‌گرداند تا سهمی از جایزه‌ی بلاک به دست آورد.

ائتلاف استخراج همتا-به-همتا (P2Pool)

ائتلاف‌های مدیریت‌شده با احتمال تقلب رهبر ائتلاف، که می‌تواند از کار ائتلاف برای خرج- دوباره‌ی تراکنش‌ها یا نامعتبر کردن بلاک استفاده کند، روبرو هستند (برای توضیحات بیشتر، ادامه‌ی همین فصل را ببینید). علاوه بر آن، فعالیت کل ائتلاف به یک سرویس‌دهنده‌ی مرکزی که همیشه احتمال خرابی آن وجود دارد، وابسته است. اگر این سرویس‌دهنده از کار بیفتد یا در اثر حمله‌ی DoS به شدت کُند شود، معدنچیان عضو ائتلاف قادر به استخراج بیت‌کوین نخواهند بود. در سال ۲۰۱۱، برای برطرف کردن مشکلات استخراج ائتلافی متمرکز، یک روش استخراج ائتلافی جدید پیشنهاد و پیاده‌سازی شد: ائتلاف استخراج همتا-به-همتا (peer-to-peer mining pool) بدون نیاز به راهبر مرکزی.

در P2Pool وظایف و کارکردهای سرویس‌دهنده‌ی ائتلاف توزیع شده است، و یک سیستم شبه- بلاک‌چین موازی موسوم به زنجیره‌ی سهام (share chain) پیاده‌سازی می‌شود. زنجیره‌ی سهام یک بلاک‌چین است که با دشواری کمتری نسبت به بلاک‌چین بیت‌کوین کار می‌کند. زنجیره‌ی سهام به معدنچیان اجازه می‌دهد با استخراج سهم در یک زنجیره‌ی سهام با آهنگ یک بلاک سهم در هر ۳۰ ثانیه در یک ائتلاف غیرمتمرکز به مشارکت بپردازند.

هر بلاک سهم که در زنجیره‌ی سهام ثبت می‌شود، سهمیه‌ای متناسب برای معدنچی استخراج‌کننده‌ی آن بلاک سهم به ارمغان می‌آورد؛ و این بلاک‌های سهم در زنجیره روی یکدیگر انباشته می‌شوند. همچنین وقتی یکی از این بلاک‌های سهم هدف دشواری شبکه‌ی بیت‌کوین را برآورده می‌کنند، این بلاک در شبکه‌ی بیت‌کوین منتشر شده و به بلاک‌چین بیت‌کوین اضافه می‌شود، و جایزه‌ی استخراج بلاک برنده متناسب با سهم هر معدنچی بین اعضای گروه تقسیم خواهد شد. در این روش، به جای استفاده از یک سرویس‌دهنده‌ی مرکزی برای ثبت و پیگیری میزان مشارکت و سهمیه‌ی هر معدنچی، زنجیره‌ی سهام به تمام معدنچیان اجازه می‌دهد تا با ساز و کاری غیرمتمرکز (شبیه ساز و کار اجماع در بلاک‌چین بیت‌کوین) سهم هر فرد را محاسبه و پرداخت کنند.

استخراج P2Pool پیچیده‌تر از استخراج ائتلافی است، چون هر معدنچی باید سخت‌افزار اختصاصی با فضای دیسک، حافظه و پهنای باند اینترنت کافی برای پشتیبانی از یک گره کامل بیت‌کوین و نرم‌افزار گره P2Pool فراهم کند. معدنچی P2Pool تجهیزات استخراج خود را به این گره P2Pool (که با ارسال الگوی بلاک نامزد به این دستگاه‌ها، کارکرد سرویس‌دهنده‌ی ائتلاف را شبیه‌سازی خواهد کرد) متصل می‌کند. در P2Pool، هر معدنچی به طور مستقل (درست مثل یک معدنچی تکرو) بلاک نامزد خاص خود را می‌سازد و تراکُنش‌ها را در آن تجمیع می‌کند، ولی استخراج این بلاک را به صورت اشتراکی و در زنجیره‌ی سهام انجام می‌دهد. استخراج P2Pool یک رویکرد ترکیبی با مزیت جایزه‌ی بیشتر نسبت به استخراج مستقل است، که در ضمن در آن خبری از کنترل شدید راهبر ائتلاف (مانند ائتلاف‌های استخراج مدیریت‌شده) نیست.

هر چند روش استخراج P2Pool تمرکز قدرت در دست راهبر ائتلاف را کاهش می‌دهد، ولی همچنان در مقابل «حمله‌ی ۵۱٪» به خود زنجیره‌ی سهام آسیب‌پذیر است. در حقیقت، گسترش محبوبیت و کاربرد P2Pool مشکل «حمله‌ی ۵۱٪» به خود بیت‌کوین را حل نمی‌کند، ولی با ایجاد یک زیست‌بوم استخراج متنوع باعث مستحکم‌تر شدن کل سیستم بیت‌کوین می‌شود.

انواع حمله‌ی اجماع

ساز و کار اجماع بیت‌کوین (دستکم از لحاظ نظری) در مقابل حمله‌ی معدنچیان یا ائتلاف‌ها برای استفاده از قدرت دَرهم‌سازی آن در جهت اهداف متقلبانه یا مخرّب آسیب‌پذیر است. همان طور که دیدیم، ساز و کار اجماع به این بستگی دارد که اکثریت معدنچیان شرافتمندانه و خالی از خودخواهی عمل کنند. با این حال، اگر یک معدنچی یا یک ائتلاف از معدنچیان بتوانند سهمی قابل‌توجه از قدرت استخراج (دَرهم‌سازی) را در اختیار بگیرند، می‌توانند به ساز و کار اجماع بیت‌کوین حمله کرده و امنیت و دسترس‌پذیری آن را مختل کنند.

شایان ذکر است که حمله‌های اجماع فقط می‌توانند اجماع آینده، یا در بهترین حالت اجماع گذشته‌ی نزدیک (حداکثر چند ده بلاک) را تحت تأثیر قرار دهند. دفتر کل بیت‌کوین (بلاک‌چین) با گذشت زمان تغییرناپذیرتر می‌شود. هر چند از لحاظ نظری یک انشعاب می‌تواند به هر عمقی برسد، ولی در عمل قدرت پردازش مورد نیاز برای ایجاد یک انشعاب خیلی عمیق چنان زیاد است که بلاک‌های قدیمی را عملاً تغییرناپذیر می‌کند. حمله‌های اجماع همچنین تأثیری بر امنیت کلیدهای خصوصی و الگوریتم امضای دیجیتال (ECDSA) ندارند. یک حمله‌ی اجماع نمی‌تواند بیت‌کوین سرقت کند، سکه‌های بیت‌کوین را بدون امضا خرج کند، بیت‌کوین را به آدرس‌های دیگر حواله کند، یا حتی تراکُنش‌های قدیمی را تغییر داده یا مالکیت سکه‌های بیت‌کوین را عوض کند. حمله‌های اجماع فقط جدیدترین بلاک‌ها را تحت تأثیر قرار می‌دهند و باعث اختلال DoS (قطع سرویس) در تولید بلاک‌های آینده می‌شوند.

یکی از معروف‌ترین سناریوهای حمله به ساز و کار اجماع «حمله‌ی ۵۱٪» نام دارد. در این سناریو، ائتلافی از معدنچیان با در اختیار گرفتن اکثریت (۵۱٪) قدرت دَرهم‌سازی شبکه‌ی بیت‌کوین، برای حمله به آن تبانی می‌کنند. با داشتن توانایی استخراج اکثریت بلاک‌ها، حمله‌کنندگان می‌توانند انشعاب‌های عمدی در بلاک‌چین ایجاد کرده و تراکُنش‌ها را بیش از یک بار خرج کنند، یا دست به حمله‌های DoS علیه تراکُنش‌ها یا آدرس‌های خاص بزنند.

در یک حمله‌ی انشعاب/خرج-دوباره، حمله‌کننده می‌تواند با ایجاد انشعاب در زیر بلاک‌های قبلاً تأییدشده و همراستا کردن شبکه روی یک زنجیره‌ی متفاوت، این بلاک‌ها را نامعتبر کند. چنین حمله‌کننده‌ای با داشتن قدرت درهم‌سازی کافی می‌تواند شش بلاک متوالی (یا بیشتر) را نامعتبر کرده و تراکنش‌هایی را که تغییرناپذیر تلقی می‌شوند (یعنی شش تأییدیه دارند)، از اعتبار ساقط کند. البته توجه داشته باشید که این حمله‌کننده فقط می‌تواند تراکنش‌های خودش را بیش از یک بار خرج کند، چون فقط برای این تراکنش‌ها است که می‌تواند امضای معتبر تولید کند. همچنین خرج-دوباره‌ی تراکنش‌ها فقط زمانی سودآور است که حمله‌کننده بعد از نامعتبر کردن یک تراکنش، کالا یا خدماتی را که در قبال آن دریافت کرده، از دست ندهد.

اجازه دهید یک مثال عملی از «حمله‌ی ۵۱٪» را بررسی کنیم. در فصل اول تراکنشی بین آلیس و باب برای خرید یک فنجان قهوه دیدیم. باب، صاحب مغازه‌ی قهوه‌فروشی، چندان اصراری برای تأییدیه‌ی سریع تراکنش فروش یک فنجان قهوه (یعنی بلافاصله در بلاک بعدی) ندارد چون از نظر او خطر خرج-دوباره‌ی تراکنشی با این مبلغ پایین (یک فنجان قهوه) در مقایسه با نارضایتی مشتری ناچیز است. این وضعیت درست مثل فروشگاه‌هایی است که برای خریدهای زیر ۲۵ دلار با کارت اعتباری منتظر امضای مشتری و دریافت تأییدیه‌ی بانک نمی‌مانند، چون خطر از دست دادن این مبلغ در مقایسه با هزینه‌ی تأخیر ناشی از دریافت تأییدیه‌ی بانک و نارضایتی مشتری قابل چشم‌پوشی است. از سوی مقابل، فروش اقلام گران‌تر در ازای دریافت بیت‌کوین با خطر حمله‌ی خرج-دوباره مواجه است، چون خریدار می‌تواند بلافاصله یک تراکنش دیگر با همان UTXO انجام داده و پرداخت قبلی خود را لغو کند. حمله‌ی خرج-دوباره می‌تواند به دو صورت اتفاق بیفتد: قبل از تأیید شدن یک تراکنش، یا اگر حمله‌کننده توانایی ایجاد انشعاب عمیق در بلاک‌چین و نامعتبر کردن چندین بلاک را داشته باشد. یک «حمله‌ی ۵۱٪» به حمله‌کننده اجازه می‌دهد تراکنش‌های خرج‌شده‌ی قبلی خود را دوباره در یک زنجیره‌ی جدید خرج کرده و تراکنش‌های زنجیره‌ی قدیمی را خنثی کند.

این بار از افراد دیگری که در فصل ۱ معرفی کردیم، برای نمایش یک «حمله‌ی ۵۱٪» دیگر استفاده کنیم. مالوری (حمله‌کننده‌ی بدطینت) به نگارخانه‌ی کارول می‌رود و یک تابلوی نقاشی سه‌تکه‌ی زیبا به نام «آتش بزرگ» (که ساتوشی ناکاموتو را در قالب پروانه به تصویر کشیده) از او می‌خرد. کارول که روی این نقاشی ۲۵۰ هزار دلار قیمت گذاشته، قبول می‌کند بهای آن را به بیت‌کوین از مالوری دریافت کند. اما کارول به جای آن که صبر کند تا این تراکنش در شش بلاک متوالی تأیید شود، بعد از دیدن اولین تأییدیه، تابلو را به مالوری تحویل می‌دهد. مالوری همکاری به نام پاول دارد که یک ائتلاف استخراج بزرگ را اداره می‌کند، و بلافاصله بعد از تأیید تراکنش مالوری در اولین بلاک و تحویل گرفتن تابلوی نقاشی، یک «حمله‌ی ۵۱٪» را ترتیب می‌دهد. پاول ترتیبی می‌دهد تا ائتلاف او یک بلاک با همان ارتفاع بلاکی که حاوی تراکنش مالوری است، استخراج کند و به جای تراکنش پرداخت مالوری به کارول، تراکنش جدیدی در آن قرار می‌دهد که همان ورودی را دوباره خرج می‌کند. این تراکنش همان UTXO را خرج کرده و تمامی مبلغ آن را به جای پرداخت به کارول به کیف‌پول مالوری برمی‌گرداند، و در واقع به مالوری اجازه می‌دهد کل این مبلغ را برای خود نگه دارد. پاول سپس ائتلاف استخراج خود را طوری هدایت می‌کند که یک بلاک دیگر استخراج کنند، و زنجیره‌ای می‌سازد که حاوی تراکنش خرج-دوباره‌ی مالوری بوده و در ضمن طول آن از زنجیره‌ی اصلی بیشتر است؛ در واقع، پاول با این عمل یک انشعاب زیر بلاک حاوی تراکنش مالوری به وجود می‌آورد.

وقتی کمی بعد این انشعاب در بلاک‌چین به نفع زنجیره‌ی جدید (که بلندتر است) حل شود، تراکنش خرج-دوباره (پرداخت UTXO به کیف‌پول مالوری) جایگزین تراکنش اولیه (پرداخت UTXO به کیف‌پول کارول) خواهد شد؛ کارول یک تابلوی نقاشی گرانقیمت از دست داده و هیچ بیت‌کوینی هم دریافت نکرده است. در تمام این ماجرا، نقش گروه استخراج پاول در این حمله‌ی خرج-دوباره از نظرها مخفی می‌ماند، چون آنها در ظاهر فقط یک ائتلاف استخراج معمولی هستند که نمی‌توانند بر همه‌ی تراکنش‌ها یا بلاک‌ها نظارت دقیق داشته باشند.

برای حفاظت در مقابل این نوع حمله، کسانی که کالاهای گرانقیمت می‌فروشند، باید قبل از تحویل کالا به خریدار حداقل شش تأییدیه (بلاک) منتظر بمانند. همچنین معامله‌گرانی که باید از یک حساب تضمینی چندامضایی استفاده کنند، باید بعد از ایجاد این حساب چندین بلاک (تأییدیه) صبر کنند. هر چه تعداد تأییدیه‌ها بیشتر باشد، نامعتبر کردن یک تراکُنش با استفاده از «حمله‌ی ۵۱٪» دشوارتر خواهد شد. برای اجناس بسیار گرانقیمت یا معاملات خیلی بزرگ توصیه می‌شود فروشنده حداقل ۲۴ ساعت (که تقریباً معادل ۱۴۴ تأییدیه است) منتظر بماند.

یکی دیگر از کاربردهای حمله‌ی اجماع (غیر از حمله‌ی خرج-دوباره) قطع کردن دسترسی یک آدرس [فرد] خاص به شبکه‌ی بیت‌کوین است. حمله‌کننده‌ای که بخش اکثر قدرت دَرهم‌سازی شبکه را در اختیار دارد، می‌تواند به سادگی تراکُنش‌هایی را که به یک آدرس خاص انجام شده‌اند، نادیده بگیرد. حتی اگر این تراکُنش‌ها در بلاک‌هایی که توسط معدنچیان دیگر استخراج شده‌اند، قرار گرفته باشند، حمله‌کننده می‌تواند با استخراج مجدد همان بلاک و ایجاد یک انشعاب، آن تراکُنش‌ها را از بلاک‌چین بیرون بیندازد. اگر حمله‌کننده از قدرت دَرهم‌سازی کافی برخوردار باشد، این نوع حمله می‌تواند باعث قطع ارتباط یک یا چند آدرس خاص با شبکه‌ی بیت‌کوین شود.

باوجود نام آن، برای اجرای موفقیت‌آمیز یک «حمله‌ی ۵۱٪» نیازی نیست حمله‌کننده واقعاً ۵۱٪ قدرت دَرهم‌سازی شبکه‌ی بیت‌کوین را در کنترل خود داشته باشد؛ در حقیقت، این حمله را با درصد کوچکتری از قدرت دَرهم‌سازی شبکه هم می‌توان انجام داد. آستانه‌ی ۵۱٪ فقط مقداری است که موفقیت این حمله را تقریباً تضمین می‌کند. حمله‌ی اجماع اساساً یک زورآزمایی برای استخراج بلاک بعدی است که در آن احتمال برنده شدن ائتلاف «قویتر» بیشتر است. با قدرت دَرهم‌سازی کمتر، احتمال موفقیت هم کاهش می‌یابد، چون احتمال استخراج بلاک بعدی توسط معدنچیان دیگر (که در این حمله شرکت ندارند) بیشتر می‌شود. در واقع، هر چه قدرت دَرهم‌سازی یک حمله‌کننده بیشتر باشد، انشعاب بلندتری می‌تواند ایجاد کند، و بلاک‌های قبلاً استخراج‌شده‌ی بیشتری را می‌تواند نامعتبر کرده، یا بلاک‌های بیشتری را در آینده کنترل کند. مدل‌سازی آماری گروه‌هایی که روی امنیت شبکه تحقیق می‌کنند، نشان داده است که با کنترل حداقل ۳۰٪ از قدرت دَرهم‌سازی شبکه می‌توان انواع مختلفی از حمله‌ی اجماع را با احتمال موفقیت نسبتاً بالا انجام داد.

افزایش عظیم قدرت دَرهم‌سازی شبکه‌ی بیت‌کوین آسیب‌پذیری آن در مقابل حمله‌ی معدنچی‌های منفرد را به شدت کاهش داده است. در واقع، هیچ راهی وجود ندارد که یک معدنچی تکرو بتواند مقداری بیش از درصد کوچکی از قدرت دَرهم‌سازی کل شبکه را در کنترل خود داشته باشد. با این حال، تمرکز قدرت دَرهم‌سازی در ائتلاف‌های استخراج خطر جدیدی را ایجاد کرده است: حمله‌هایی که از جانب راهبران ائتلاف‌ها برای کسب منافع بیشتر طراحی می‌شوند. همان طور که دیدیم، ایجاد بلاک نامزد و انتخاب تراکُنش‌هایی که در این بلاک قرار می‌گیرند، از اختیارات راهبر ائتلاف استخراج مدیریت‌شده است. این ویژگی به راهبر ائتلاف اجازه می‌دهد برخی تراکُنش‌ها را نادیده بگیرد (حمله‌ی DoS)، یا آنها را دوباره خرج کند (حمله‌ی خرج-دوباره). اگر راهبر ائتلاف بتواند این قبیل حمله‌های اجماع را به صورت محدود و با ظرافت انجام دهد، به احتمال زیاد می‌تواند بدون این که کسی متوجه شود، سود زیادی نصیب خود کند.

با این حال، همه‌ی حمله‌های اجماع با انگیزه‌ی کسب منفعت انجام نمی‌شوند. شاید نیت یک حمله‌کننده فقط تخریب شبکه‌ی بیت‌کوین باشد، بدون این که نفعی از این تخریب ببرد. حمله‌ای که بتواند کل شبکه‌ی بیت‌کوین را به زانو در آورد، به حجم عظیمی از سرمایه‌گذاری و تجهیزات و برنامه‌ریزی مخفیانه نیاز دارد، ولی احتمال آن هست که برخی کشورها چنین هدفی را دنبال کنند. از طرف دیگر، اگر یک حمله‌کننده‌ی مصمم همزمان تجهیزات دَرهم‌سازی انبوهی گرد آورد، با راهبران ائتلاف‌های استخراج مدیریت‌شده زد و بند کند، و دست به حمله‌ی DoS به ائتلاف‌های دیگر (که موفق به خریدن راهبران آنها نشده) بزند، شاید بتواند ضربه‌ی سختی به ساز و کار اجماع بیت‌کوین وارد آورد. همه‌ی این سناریوها از لحاظ نظری محتمل هستند، ولی با رشد انفجاری قدرت دَرهم‌سازی کل شبکه‌ی بیت‌کوین هر روز بیشتر از قبل غیرعملی (ناممکن) می‌شوند.

یک حمله‌ی اجماع جدی بی‌تردید در کوتاه‌مدت باعث از بین رفتن اعتماد به شبکه‌ی بیت‌کوین خواهد شد، و حتی می‌تواند به سقوط قیمت بیت‌کوین منجر شود. با این حال، شبکه‌ی بیت‌کوین و نرم‌افزارهای آن به طور پیوسته در حال تکامل هستند، بنابراین چنین حمله‌ای بلافاصله با مقابله‌ی جدی جامعه‌ی بیت‌کوین روبرو خواهد شد که بیت‌کوین را قویتر، مستحکم‌تر، و مخفی‌تر از قبل می‌کند.

تغییر قواعد اجماع

قواعد اجماع (consensus rules) اعتبار تراکُنش‌ها و بلاک‌ها را تعیین می‌کنند. این قواعد مبنایی برای همکاری بین گره‌های بیت‌کوین بوده و مسئول همگرایی نماهای محلی از بلاک‌چین آن در یک بلاک‌چین واحد و همسان در سرتاسر شبکه هستند.

هر چند قواعد اجماع در کوتاه‌مدت ثابت هستند و بایستی در میان تمام گره‌های شبکه همسان باشند، ولی در بلندمدت می‌توانند تغییر کنند. برای تکامل و توسعه‌ی سیستم بیت‌کوین، این قواعد باید در کنار معرفی ویژگی‌های جدید، رفع نقاط ضعف، یا ارتقای سیستم به‌روز شوند. با این حال، بر خلاف فرآیند توسعه در نرم‌افزارهای سنتی، ارتقای یک سیستم مبتنی بر اجماع بسیار دشوارتر است و نیاز به هماهنگی بین تمامی شرکت‌کنندگان دارد.

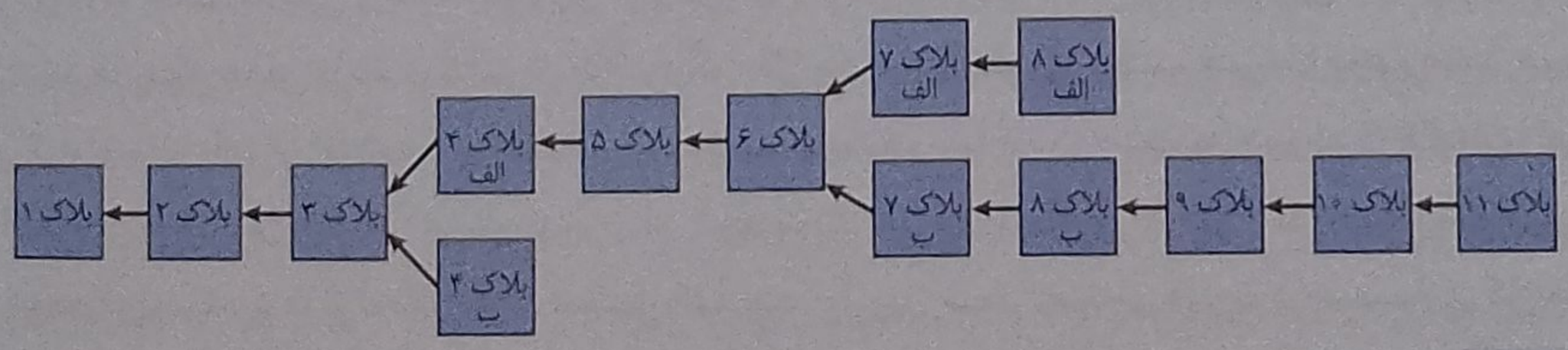
انشعاب سخت

در قسمت‌های قبل دیدیم که چگونه شبکه‌ی بیت‌کوین می‌تواند برای مدتی کوتاه واگرا شود، به طوری که دو بخش از شبکه برای دقایقی دو بلاک‌چین متفاوت را دنبال کنند. گفتیم که چرا در اثر عملکرد طبیعی شبکه چنین اتفاقی می‌افتد، و چگونه شبکه‌ی بیت‌کوین بعد از استخراج یک یا (حداکثر) دو بلاک روی یک بلاک‌چین مشترک همراستا می‌شود.

موقعیت دیگری که شبکه‌ی بیت‌کوین به دو زنجیره‌ی متفاوت واگرا می‌شود، تغییر در قواعد اجماع است. بعد از این نوع انشعاب که به آن **انشعاب سخت** (hard fork) گفته می‌شود، شبکه‌ی بیت‌کوین دیگر در یک زنجیره‌ی واحد همگرا نخواهد شد، بلکه به جای آن دو زنجیره‌ی مختلف ایجاد می‌شود که به طور مستقل پیش می‌روند. انشعاب سخت زمانی روی می‌دهد که بخشی از شبکه بر اساس مجموعه‌ای متفاوت از قواعد اجماع با بقیه‌ی شبکه عمل کند، اتفاقی که می‌تواند به خاطر وجود یک باگ نرم‌افزاری یا تغییر عمدی در پیاده‌سازی قواعد اجماع پیش آید.

از انشعاب سخت می‌توان برای تغییر دادن قواعد اجماع استفاده کرد، ولی این کار مستلزم هماهنگی بین تمامی اعضای سیستم است. هر گرهی که خود را با قواعد اجماع جدید وفق ندهد، دیگر نمی‌تواند در ساز و کار اجماع مشارکت کند و (درست در لحظه‌ی انشعاب سخت) به یک زنجیره‌ی جداگانه تبعید می‌شود. بنابراین، تغییر حاصل از یک انشعاب سخت را می‌توان تغییری دانست که **سازگاری پیش‌رو** (forward compatibility) ندارد، و سیستم‌هایی که خود را ارتقا نداده‌اند دیگر قادر به پردازش قواعد اجماع جدید نخواهند بود.

اجازه دهید ساز و کار انشعاب سخت را با یک مثال مشخص بررسی کنیم. در شکل ۱۰-۹ یک بلاک‌چین با دو انشعاب مشاهده می‌کنید. اولین انشعاب در ارتفاع بلاک ۴ رُخ می‌دهد. این یک انشعاب خود به خود است که در قسمت‌های قبل با آن آشنا شدیم. با استخراج بلاک ۵ این انشعاب برطرف شده و شبکه دوباره روی یک زنجیره همراستا می‌شود.



شکل ۱۰-۹ یک بلاک‌چین با دو انشعاب.

با این حال، کمی بعد یک انشعاب سخت در ارتفاع بلاک ۶ رُخ می‌دهد. فرض کنید دلیل این انشعاب منتشر شدن یک پیاده‌سازی جدید با تغییراتی در قواعد اجماع بوده است. از بلاک ۷ به بعد، معدنچیانی که این پیاده‌سازی جدید را در گره‌های خود اجرا می‌کنند، به جای امضای دیجیتال مبتنی بر ECDSA، نوع جدید و متفاوتی از امضای دیجیتال که آن را (مثلاً) «کلوچه» می‌نامیم، قبول خواهند کرد. چند لحظه بعد، یکی از گره‌هایی که پیاده‌سازی جدید را نصب و اجرا کرده، تراکُنشی حاوی یک امضای «کلوچه» تولید می‌کند، و یک معدنچی (که آن هم پیاده‌سازی جدید را نصب و اجرا کرده) بلاک ۷ ب را که حاوی این تراکُنش است، استخراج می‌کند.

گره‌هایی که نرم‌افزار خود را ارتقا نداده‌اند، قادر به اعتبارسنجی امضاهای «کلوچه» نیستند، و در نتیجه نمی‌توانند بلاک ۷ ب را پردازش (استخراج) کنند. از دیدگاه آنها بلاک‌های حاوی تراکُنش‌هایی که این امضا در آنها به کار گرفته شده، مانند بلاک ۷ب، نامعتبر هستند چون این گره‌ها اعتبارسنجی تراکُنش‌ها را همچنان بر مبنای قواعد اجماع قدیمی انجام می‌دهند. در نتیجه این قبیل تراکُنش‌ها و بلاک‌ها را رد کرده و آنها را منتشر نخواهند کرد. معدنچیانی که از قواعد اجماع قدیمی پیروی می‌کنند، بلاک ۷ ب را نپذیرفته و به استخراج بلاک‌هایی که فرزند بلاک ۶ هستند، ادامه خواهند داد. در حقیقت، اگر تمامی گره‌های همسایه‌ی متصل به یک معدنچی از ویرایش قدیمی قواعد اجماع استفاده کنند، این معدنچی حتی ممکن است بلاک ۷ ب را دریافت نکند، چون گره‌های همسایه‌ی او بلاک ۷ ب را دور می‌اندازند و آن را منتشر نمی‌کنند. اما این گره‌ها قادر به پردازش بلاک ۷ الف هستند، بلاکی که تحت قواعد اجماع قدیمی تولید شده و حاوی تراکُنش‌های دارای امضای «کلوچه» نیست.

واگرایی دو زنجیره از این نقطه ادامه می‌یابد. معدنچیان زنجیره‌ی «ب» به دریافت، پذیرش و استخراج تراکُنش‌های حاوی امضای «کلوچه» ادامه خواهند داد، در حالی که معدنچیان زنجیره‌ی «الف» همچنان این قبیل تراکُنش‌ها را نامعتبر تلقی کرده و آنها را رد خواهند کرد. حتی اگر بلاک ۸ ب حاوی هیچ تراکُنش دارای امضای «کلوچه» نباشد، باز هم معدنچیان زنجیره‌ی «الف» آن را پردازش نخواهند کرد، چون این گره‌ها بلاک ۷ ب (مادر بلاک ۸ ب) را به رسمیت نمی‌شناسد و از دیدگاه آنها بلاک ۸ ب یک بلاک یتیم است.

انشعاب سخت: نرم‌افزار، شبکه، استخراج، و زنجیره

واژه‌ی «انشعاب» برای برنامه‌نویسان معنای دیگری نیز دارد، که باعث ابهام بیشتر در اصطلاح «انشعاب سخت» می‌شود. در دنیای نرم‌افزارهای منبع-باز، یک انشعاب زمانی رُخ می‌دهد که گروهی از برنامه‌نویسان راه خود را از سایرین جدا کرده و مسیر جدیدی برای توسعه و پیاده‌سازی یک پروژه‌ی منبع-باز در پیش می‌گیرند. تا اینجا دو وضعیت را که منجر به انشعاب سخت می‌شوند، بررسی کردیم: یک باگ نرم‌افزاری، یا تغییر عمدی قواعد اجماع. در مورد دوم (تغییر عمدی قواعد اجماع)، انشعاب سخت همیشه بعد از یک انشعاب نرم‌افزاری می‌آید. در نتیجه، برای ایجاد یک انشعاب سخت از نوع دوم، لازم است ابتدا قواعد اجماع جدید پیاده‌سازی و [از طرف همگان یا بخش بزرگی از شبکه] پذیرفته شده و رسماً منتشر شوند.

چند نمونه از انشعاب‌های نرم‌افزاری که برای تغییر قواعد اجماع تلاش کرده‌اند، عبارتند از «بیت‌کوین XT»، «بیت‌کوین کلاسیک»، و جدیدترین آنها «بیت‌کوین نامحدود». با این حال، هیچ یک از این انشعاب‌های نرم‌افزاری منجر به یک انشعاب سخت نشده‌اند. با آن که انشعاب نرم‌افزاری پیش‌شرط لازم انشعاب سخت است، ولی به تنهایی برای ایجاد یک انشعاب سخت کافی نیست. برای این که یک انشعاب سخت اتفاق بیفتد، پیاده‌سازی‌های رقیب بایستی توسط جامعه‌ی بیت‌کوین [معدنچیان، کیف‌پول‌ها و گره‌های بینابینی] پذیرفته شده و قواعد اجماع جدید فعال شوند. برعکس، تعداد زیادی پیاده‌سازی مختلف از هسته‌ی بیت‌کوین، و حتی انشعاب‌های نرم‌افزاری، وجود دارند که قواعد اجماع را تغییر نمی‌دهند، و (اگر باگ مهمی در آنها وجود نداشته باشد) می‌توانند بدون ایجاد یک انشعاب سخت روی شبکه همزیستی و همکاری متقابل داشته باشند.

قواعد اجماع ممکن است در اعتبارسنجی تراکُنش‌ها یا بلاک‌ها تفاوت‌های آشکار و صریحی با یکدیگر داشته باشند. وجود تفاوت‌های ظریف و پنهان، از نظر پیاده‌سازی این قواعد در زمینه‌ی اسکریپت‌های بیت‌کوین یا کارکردهای رمزنگاری مانند امضای دیجیتال، نیز دور از تصور نیست. سرانجام، شاید به خاطر الزامات ضمنی ناشی از محدودیت‌های سیستم یا جزئیات پیاده‌سازی، تفاوت‌های غیرمنتظره‌ای در قواعد اجماع بروز کند. برای نمونه، وقتی هسته‌ی بیت‌کوین از ویرایش ۰٫۷ به ۰٫۸ ارتقا داده شد، به خاطر وجود یک محدودیت در پیاده‌سازی پایگاه داده‌ی برکلی (که برای ذخیره‌سازی بلاک‌ها به کار می‌رفت)، یک انشعاب سخت اتفاق افتاد.

از نظر مفهومی، یک انشعاب سخت در چهار مرحله اتفاق می‌افتد: انشعاب در نرم‌افزار، انشعاب در شبکه، انشعاب در [فرآیند] استخراج، و انشعاب در زنجیره. فرآیند انشعاب سخت با ورود یک پیاده‌سازی جدید از مشتری [هسته‌ی بیت‌کوین] که در آن قواعد اجماع (توسط برنامه‌نویسان) تغییر داده شده، آغاز می‌شود.

با توزیع [انتشار] این پیاده‌سازی جدید در شبکه، بخشی از معدنچیان، کیف‌پول‌ها و گره‌های بینابینی آن را پذیرفته و روی سیستم‌های خود اجرا می‌کنند. انشعاب حاصل به آن بستگی خواهد داشت که قواعد اجماع جدید درباره‌ی کدام جنبه از سیستم بیت‌کوین است: بلاک‌ها، تراکُنش‌ها، یا دیگر جنبه‌های سیستم. اگر قواعد اجماع جدید به تراکُنش‌ها مربوط باشد، آنگاه یک کیف‌پول که مسئول ایجاد تراکُنش است، می‌تواند با پیروی از قواعد جدید باعث انشعاب در شبکه شود، که بعد از استخراج این تراکُنش (در یک بلاک) انشعاب سخت به وجود خواهد آورد. اگر قواعد جدید مربوط به بلاک‌ها باشد، آنگاه انشعاب سخت با استخراج یک بلاک (تحت این قواعد) شروع خواهد شد.

ابتدا شبکه دچار انشعاب خواهد شد. گره‌هایی که هنوز بر اساس قواعد اجماع اولیه کار می‌کنند، تراکُنش‌ها و بلاک‌های ایجادشده با قواعد جدید را رد خواهند کرد. علاوه بر آن، این گره‌ها ارتباط خود را با گره‌هایی که این تراکُنش‌ها و بلاک‌های (به باور آنها) نامعتبر را منتشر می‌کنند، قطع کرده و آنها را به طور موقت تحریم خواهند کرد. در نتیجه، شبکه به دو بخش تقسیم می‌شود: گره‌های قدیمی که فقط با گره‌های قدیمی ارتباط دارند، و گره‌های جدید که فقط با گره‌های جدید ارتباط خواهند داشت. همان طور که می‌بینید، یک تراکُنش یا بلاک که بر اساس قواعد اجماع جدید ساخته شده، در مدتی کوتاه کل شبکه را در می‌نوردد و آن را به دو پاره تقسیم می‌کند.

همین که یک معدنچی با استفاده از قواعد جدید یک بلاک را استخراج کند، قدرت پردازش (دَرهم‌سازی) و زنجیره نیز دچار انشعاب خواهند شد. معدنچیان جدید استخراج را بر مبنای این بلاک جدید انجام می‌دهند، در حالی که معدنچیان قدیمی به استخراج یک زنجیره‌ی جداگانه بر اساس قواعد قدیمی ادامه خواهند داد. از آنجا که شبکه از قبل به دو اردوگاه متباین تقسیم شده است، معدنچیانی که بر اساس قواعد اجماع متفاوت عمل می‌کنند، بلاک‌های یکدیگر را دریافت نخواهند کرد.

واگرایی معدنچیان و دشواری

با تقسیم معدنچیان به دو زنجیره (اردوگاه) مختلف، قدرت دَرهم‌سازی شبکه بین این زنجیره‌ها تقسیم می‌شود. تقسیم قدرت استخراج بین دو زنجیره معمولاً به نسبت مساوی اتفاق نمی‌افتد، و در واقع می‌تواند هر نسبتی داشته باشد. شاید قواعد اجماع جدید طرفداران کمی پیدا کند، یا این که ممکن است اکثریت شبکه با آن همراه شوند. به عنوان مثال، اجازه دهید فرض کنیم قدرت استخراج شبکه به نسبت ۲۰٪ - ۸۰٪ به نفع قواعد اجماع جدید تقسیم شده است. همچنین فرض کنید این انشعاب درست بعد از یک دوره‌ی بازتنظیم هدف دشواری اتفاق افتاده است.

اکنون این دو زنجیره یک هدف دشواری واحد را از زنجیره‌ی قدیمی ارث برده‌اند. زنجیره‌ی مبتنی بر قواعد اجماع جدید ۸۰٪ قدرت استخراج (دَرهم‌سازی) موجود شبکه را با خود دارد. از دیدگاه این زنجیره، قدرت استخراج شبکه ناگهان نسبت

به دوره‌ی قبل ۲۰٪ افت کرده است و اکنون بلاک‌های جدید (به دلیل ۲۰٪ کاهش در قدرت دَرهم‌سازی این شاخه از شبکه) به طور میانگین در فواصل ۱۲-دقیقه‌ای استخراج می‌شوند. اگر معدنچی جدیدی به این شاخه اضافه نشده و قدرت استخراج آن در اثر عوامل خارجی افزایش نیابد، این آهنگ استخراج بلاک‌های جدید تا ۲۰۱۶ بلاک بعد، تقریباً ۲۴,۱۹۲ دقیقه (با سرعت ۱ بلاک در ۱۲ دقیقه) یا ۱۶٫۸ روز، حفظ خواهد شد. بعد از ۱۶٫۸ روز، دوره‌ی بازتنظیم دشواری بعدی فرا می‌رسد و هدف دشواری ۲۰٪ کاهش داده می‌شود تا آهنگ استخراج بلاک جدید به ۱۰ دقیقه بازگردد.

اوضاع برای زنجیره‌ی اقلیت که همچنان به قواعد اجماع قدیمی پایبند است و اکنون فقط ۲۰٪ از قدرت استخراج شبکه را در اختیار دارد، از این هم سخت‌تر است، چون در این شاخه از شبکه قدرت استخراج ۸۰٪ نسبت به دوره‌ی قبل افت کرده، و بلاک‌های جدید اکنون به طور میانگین هر ۵۰ دقیقه استخراج می‌شوند. اما هدف دشواری این شبکه تا ۲۰۱۶ بلاک بعد، که استخراج آنها تقریباً ۱۰۰,۸۰۰ دقیقه (با سرعت ۱ بلاک در ۵۰ دقیقه) یا حدود ۱۰ هفته طول می‌کشد، تغییر نخواهد کرد. با فرض این که ظرفیت تراکُنش‌های هر بلاک ثابت بماند، این وضعیت باعث افت ۸۰٪ در ظرفیت پردازش تراکُنش در این شاخه از شبکه نیز خواهد شد، چون (تقریباً) در هر ساعت فقط یک بلاک برای تجمیع تراکُنش‌ها و استخراج می‌تواند ساخته شود.

انشعاب سخت ستیزه‌جویانه

زمان حاضر طلیعه‌ی برنامه‌نویسی اجماع است. درست همان طور که جنبش منبع-باز با ایجاد روش‌ها، ابزارها و انجمن‌های جدید دنیای نرم‌افزار را دگرگون کرد، برنامه‌نویسی اجماع نیز قلمرو جدیدی را در مقابل علوم کامپیوتر گشوده است. از میان جدل‌ها، تجربه‌ها و رنج‌های مسیر توسعه‌ی بیت‌کوین، امروزه مجموعه‌ای از ابزارهای برنامه‌نویسی، روش‌ها و جوامع جدید ظهور کرده است.

انشعاب سخت خالی از خطر نیست، چون یا اقلیت را مجبور می‌کند خود را با اکثریت وفق دهد (قواعد اجماع اکثریت را بپذیرند)، یا همچنان در زنجیره‌ی اقلیت بمانند (و دست و پا بزنند). از نظر بسیاری افراد، خطر تجزیه‌ی کل سیستم به دو سیستم رقیب قابل قبول نیست. در نتیجه، بسیاری از برنامه‌نویسان در استفاده از ساز و کار انشعاب سخت برای پیاده‌سازی اصلاحات قواعد اجماع تردید دارند، مگر این که کل شبکه به اتفاق آرا از آنها پشتیبانی کند. به عبارت دیگر، هر پیشنهاد انشعاب سخت که اتفاق آرای شبکه را نداشته باشد، اقدامی ستیزه‌جویانه و در جهت تجزیه‌ی شبکه قلمداد می‌شود.

مسأله‌ی انشعاب سخت در جامعه‌ی برنامه‌نویسان بیت‌کوین محل بحث و جدل فراوان بوده است، به ویژه در مورد آن دسته از قواعد اجماع که به کنترل حداکثر اندازه‌ی بلاک مربوط می‌شوند. برخی برنامه‌نویسان بیت‌کوین با هر شکلی از انشعاب سخت مخالف هستند و آن را بیش از حد خطرناک می‌بینند. از نظر عده‌ای دیگر ساز و کار انشعاب سخت ابزاری ضروری برای ارتقای قواعد اجماع به منظور اجتناب از «استقراض فنی» و گسست تمیز و قاطع از اشتباهات گذشته است. سرانجام، برخی دیگر معتقد هستند که ساز و کار انشعاب سخت باید به ندرت، با برنامه‌ریزی گسترده، و فقط به اتفاق آرا مورد استفاده قرار گیرد.

البته روش‌ها و تمهیداتی نیز برای مقابله با خطرات انشعاب سخت اندیشیده شده‌اند. در قسمت آینده به برخی از این تمهیدات، مانند انشعاب نرم، و روش‌هایی برای علامت‌دهی و فعال‌سازی تغییرات اجماع (BIP-34 و BIP-9)، می‌پردازیم.

انشعاب نرم

هر تغییری در قواعد اجماع منجر به یک انشعاب سخت نمی‌شود؛ فقط آن دسته از تغییرات قواعد اجماع انشعاب سخت ایجاد می‌کنند که سازگاری پیش‌رو نداشته باشند. اگر یک تغییر طوری پیاده‌سازی شود که گره‌های ارتقانیافته همچنان تراکُنش‌ها و بلاک‌های جدید را معتبر دانسته و بتوانند آنها را با همان قواعد اجماع قدیمی پردازش کنند، این تغییر باعث انشعاب نخواهد شد. برای تمایز این دسته از تغییرات قواعد اجماع با آنهایی که انشعاب سخت تولید می‌کنند، به آنها انشعاب نرم (soft fork)

گفته می‌شود، در حقیقت، یک انشعاب نرم اصلاً انشعاب نیست، بلکه فقط یک تغییر با سازگاری پیش‌رو است که اجازه می‌دهد گره‌های قدیمی به همکاری با گره‌های ارتقایافته ادامه دهند.

یک جنبه‌ی انشعاب نرم که بلافاصله در نگاه اول خود را نشان نمی‌دهد، این است که انشعاب نرم فقط می‌تواند شامل آن تغییراتی باشد که قواعد اجماع را محدودتر می‌کنند، نه آنهایی که این قواعد را بسط می‌دهند. برای حفظ سازگاری پیش‌رو، تراکُنش‌ها و بلاک‌ها باید تحت قواعد قدیمی هم معتبر بمانند، ولی عکس آن درست نیست، یعنی ممکن است یک تراکُنش یا بلاک ساخته‌شده بر اساس قواعد اجماع قدیمی با قواعد جدید معتبر تلقی نشود. در واقع، در انشعاب نرم، قواعد اجماع جدید محدودتر از قبل هستند؛ در غیر این صورت، رد شدن تراکُنش‌ها و بلاک‌ها در گره‌های قدیمی باعث بروز انشعاب سخت خواهد شد.

روش‌های متعددی برای پیاده‌سازی انشعاب نرم وجود دارد؛ در حقیقت، این اصطلاح برای بیش از یک روش کاربرد یافته است، تکنیک‌هایی که در یک چیز مشترک هستند: در یک انشعاب نرم نیازی نیست همه‌ی گره‌ها خود را ارتقا دهند، و گره‌های قدیمی از دایره‌ی اعتبارسنجی تراکُنش‌ها و بلاک‌ها بیرون رانده نمی‌شوند.

انشعاب نرم از طریق بازتعریف کُد عملیاتی NOP

تعدادی از انشعاب‌های نرم بیت‌کوین بر اساس بازتعریف کُد عملیاتی NOP پیاده‌سازی شده‌اند. اسکریپت بیت‌کوین ۱۰ کُد عملیاتی، به نام‌های NOP1 تا NOP10، را برای کاربردهای آتی کنار گذاشته است. در قواعد اجماع، حضور این کُدهای عملیاتی در یک اسکریپت *بی‌اثر* است، یعنی آنها هیچ کاری انجام نمی‌دهند. اجرای اسکریپت به سادگی بعد از کُد NOP ادامه می‌یابد، چنان که گویی این کُد اصلاً وجود نداشته است.

اما یک انشعاب نرم می‌تواند تعریف یک کُد عملیاتی NOP را تغییر داده و برای آن کارکرد جدیدی تعریف کند. برای مثال، استاندارد BIP-65 (CHECKLOCKTIMEVERIFY) کُد عملیاتی NOP2 را بازتعریف کرده است. آن دسته از مشتری‌های بیت‌کوین که BIP-65 را پیاده‌سازی کرده باشند، با دیدن کُد NOP2 در یک اسکریپت آن را به عنوان CHECKLOCKTIMEVERIFY تفسیر کرده و قاعده‌ی قفل‌زمانی مطلق را روی UTXO مربوطه (که این کُد را در اسکریپت قفل‌کننده‌ی خود دارد) اِعمال می‌کنند. این یک انشعاب نرم است، چون تراکُنشی که تحت BIP-65 معتبر است (یعنی در اسکریپت قفل‌کننده‌ی یکی از ورودی‌های خود از قفل‌زمانی CHECKLOCKTIMEVERIFY استفاده کرده)، در همه‌ی گره‌های دیگر نیز معتبر خواهد بود، زیرا آن گره‌ها به سادگی کُد NOP2 در اسکریپت قفل‌کننده‌ی این تراکُنش را نادیده می‌گیرند.

دیگر روش‌های ارتقای انشعاب نرم

بازتعریف کُدهای عملیاتی NOP ساز و کاری واضح برای ارتقای قواعد اجماع است و دقیقاً هم برای همین منظور طراحی شده اند. با این حال، در سال‌های اخیر یک ساز و کار انشعاب نرم دیگر برای نوع بسیار خاصی از تغییر در قواعد اجماع معرفی شده که متکی به کُدهای عملیاتی NOP نیست. **شاهد تفکیک‌شده** (segregated witness)، یا به اختصار **segwit**، یک تغییر معماری در ساختار تراکُنش است که اسکریپت بازکننده‌ی قفل را از داخل تراکُنش به یک ساختمان داده‌ی خارجی منتقل کرده، و در واقع آنها را از یکدیگر تفکیک می‌کند.

در ابتدا، segwit یک ارتقا از نوع انشعاب سخت تصور می‌شد، چون یک ساختار بنیادی سیستم (ساختار تراکُنش) را تغییر می‌داد. در نوامبر ۲۰۱۵، یک برنامه‌نویس که در حال کار روی هسته‌ی بیت‌کوین بود، ساز و کاری پیشنهاد کرد که با آن می‌شد segwit را به عنوان یک انشعاب نرم به کار گرفت. این ساز و کار عبارت است از نوعی تغییر در اسکریپت قفل‌کننده‌ی

UTXO ایجادشده تحت قواعد segwit، به گونه‌ای که مشتری‌های ارتقانیافته (ناآشنا با segwit) بتوانند این اسکریپت قفل‌کننده را با هر اسکریپت بازکننده‌ای دریافت (وصول) کنند. در نتیجه، برای استفاده از segwit دیگر نیازی به انتخاب بین ارتقای تمام گره‌ها یا جدا شدن از زنجیره نیست؛ به عبارت دیگر، اکنون segwit یک انشعاب نرم محسوب می‌شود.

شاید ساز و کارهای دیگری برای انشعاب نرم وجود داشته باشد که هنوز کشف نشده‌اند، و بتوان با آنها قواعد اجماع را به شیوه‌ای که سازگاری پیش‌رو داشته باشند، ارتقا داد.

مخالفان انشعاب نرم

انشعاب نرم مبتنی بر بازتعریف کُدهای عملیاتی NOP روشی نسبتاً پذیرفته‌شده است. این کُدها وقتی در اسکریپت‌های بیت‌کوین به کار گرفته می‌شوند، ارتقای قواعد اجماع را به فرآیندی بی‌دردسر و بدون نیاز به تجزیه‌ی شبکه تبدیل می‌کنند. با این حال، بسیاری از برنامه‌نویسان بیت‌کوین سایر روش‌های ارتقای انشعاب نرم را چندان بی‌خطر و قابل‌قبول نمی‌دانند. برخی از مهم‌ترین انتقادات به انشعاب نرم عبارتند از:

استقراض فنی

از آنجا که ارتقای قواعد اجماع با استفاده از انشعاب نرم از نظر فنی پیچیده‌تر از انشعاب سخت است، این انشعاب‌ها باعث ایجاد استقراض فنی می‌شوند؛ استقراض فنی (technical debt) به معنای افزایش هزینه‌ی آتی نگهداری کُدهای بیت‌کوین، به خاطر بده-بستان طراحی در گذشته، است. این پیچیدگی فنی به نوبه‌ی خود احتمال بروز باگ یا آسیب‌پذیری امنیتی را افزایش می‌دهد.

نداشتن اعتبارسنجی

وقتی یک انشعاب نرم اتفاق می‌افتد، مشتری‌های ارتقانیافته تراکُنش‌های مبتنی بر قواعد اجماع جدید را عملاً بدون هر گونه اعتبارسنجی واقعی تأیید می‌کنند. در حقیقت، مشتری‌های ارتقانیافته همچنان از قواعد اجماع قدیمی برای اعتبارسنجی تراکُنش‌ها استفاده می‌کنند، چون نسبت به قواعد اجماع جدید کور هستند. این موضوع در مورد همه‌ی روش‌های انشعاب نرم (با استفاده از کُدهای عملیاتی NOP یا segwit) صادق است.

برگشت‌ناپذیر بودن فرآیند ارتقا

از آنجا که انشعاب نرم منجر به تولید تراکُنش‌هایی می‌شود که قواعد اجماع محدودتری دارند، ارتقای قواعد اجماع با استفاده از انشعاب نرم عملاً برگشت‌ناپذیر است. اگر بخواهید یک ارتقای انشعاب نرم را بعد از فعال شدن لغو کنید، تمامی تراکُنش‌هایی که تحت قواعد اجماع جدید ایجاد شده‌اند، نامعتبر شده و ممکن است (تحت قواعد اجماع قدیمی) مبالغ آنها برای همیشه از دست برود. برای مثال، اگر یک تراکُنش CLTV تحت قواعد اجماع قدیمی پردازش شود، هیچ گونه محدودیت زمانی نخواهد داشت و می‌توان آن را در هر زمانی خرج کرد. از همین رو، منتقدان استدلال می‌کنند که اگر یک انشعاب نرم به دلیل وجود باگ مجبور به بازگشت شود، تقریباً مسجل است که مقدار زیادی بیت‌کوین نابود خواهد شد.

علامت‌دهی انشعاب نرم با ویرایش بلاک

از آنجا که انشعاب نرم به مشتری‌های ارتقانیافته اجازه می‌دهد تحت قواعد اجماع قدیمی به کار خود ادامه دهند، از یک ساز و کار «فعال‌سازی» انشعاب نرم مبتنی بر دریافت علامت «آماده» از طرف معدنچیان استفاده می‌شود: فقط وقتی اکثر معدنچیان موافقت و آمادگی خود را برای اِعمال قواعد اجماع جدید اعلام کنند، انشعاب نرم فعال خواهد شد. برای این

منظور، یک ساز و کار علامت‌دهی (signaling) پیش‌بینی شده است که به معدنچیان اجازه می‌دهد موافقت خود با یک قاعده‌ی اجماع جدید را اعلام کنند. این ساز و کار، به همراه ساز و کار فعال‌سازی (activation) انشعاب نرم، در مارس ۲۰۱۳ تحت استاندارد BIP-34 معرفی شد، و در ژوئیه ۲۰۱۶ جای خود را به ساز و کار فعال‌سازی BIP-9 داد.

علامت‌دهی و فعال‌سازی BIP-34

اولین ویرایش استاندارد BIP-34 از فیلد ویرایش بلاک استفاده می‌کرد تا به معدنچیان اجازه دهد آمادگی خود برای تغییر در یک قاعده‌ی اجماع خاص را اعلام کنند. تا قبل از BIP-34 فیلد ویرایش بلاک به طور سنتی (نه به خاطر قواعد اجماع) مقدار «۱» می‌گرفت. بنا به تعریف BIP-34، برای تغییر یک قاعده‌ی اجماع لازم بود تا فیلد پایگاه‌سکه (ورودی) تراکُنش پایگاه‌سکه حاوی ارتفاع آن بلاک باشد. همان طور که قبلاً گفتیم، در حالت معمول (قبل از BIP-34) این فیلد می‌توانست حاوی هر داده‌ای که معدنچی می‌خواست، باشد. بعد از فعال شدن BIP-34، بلاک‌های معتبر می‌بایستی حاوی یک ارتفاع بلاک مشخص در ابتدای پایگاه‌سکه بوده و با یک شماره ویرایش بزرگتر یا مساوی «۲» شناسایی می‌شدند.

برای علامت‌دهی این تغییر و فعال‌سازی BIP-34، معدنچیان باید به فیلد ویرایش بلاک مقدار «۲» می‌دادند. این کار بلافاصله باعث نامعتبر شدن بلاک‌های ویرایش «۱» نمی‌شد. اما همین که BIP-34 فعال‌سازی می‌شد، بلاک‌های ویرایش «۱» نامعتبر شده و تمامی بلاک‌های ویرایش «۲» مجبور بودند ارتفاع بلاک را در پایگاه‌سکه‌ی خود داشته باشند تا معتبر تلقی شوند.

استاندارد BIP-34 از یک ساز و کار فعال‌سازی دو-مرحله‌ای، بر اساس یک پنجره‌ی لغزان با ۱۰۰۰ بلاک، استفاده کرد. افراد (معدنچیان) آمادگی خود برای فعال‌سازی BIP-34 را با ساخت بلاک‌هایی با عدد ویرایش «۲» اعلام کردند. البته، این بلاک‌ها هنوز مجبور نبودند بر اساس قاعده‌ی اجماع جدید ارتفاع بلاک را در تراکُنش پایگاه‌سکه قرار دهند، چون این قاعده‌ی اجماع هنوز فعال نشده بود. این قاعده‌ی اجماع در دو مرحله فعال شد:

- وقتی ۷۵ درصد بلاک‌ها (۷۵۰ بلاک از آخرین ۱۰۰۰ بلاک منتشرشده در شبکه) عدد ویرایش «۲» داشتند، آنگاه بلاک‌های ویرایش «۲» بایستی حاوی ارتفاع بلاک در تراکُنش پایگاه‌سکه می‌بودند، در غیر این صورت به عنوان بلاک نامعتبر رد می‌شدند. بلاک‌های ویرایش «۱» همچنان توسط شبکه پذیرش (و پردازش) می‌شدند، و در مورد آنها نیازی به داشتن ارتفاع بلاک نبود. در این دوره قواعد اجماع قدیم و جدید همزیستی داشتند.
- وقتی تعداد بلاک‌های ویرایش «۲» به ۹۵ درصد (۹۵۰ بلاک از آخرین ۱۰۰۰ بلاک منتشرشده در شبکه) رسید، بلاک‌های ویرایش «۱» دیگر نامعتبر تلقی شده، و بلاک‌های ویرایش «۲» (درست مثل آستانه‌ی ۷۵ درصد) فقط با داشتن ارتفاع بلاک در تراکُنش پایگاه‌سکه اعتبار داشتند. از آن به بعد، تمامی بلاک‌ها باید از این قاعده‌ی اجماع پیروی می‌کردند، و بلاک‌های معتبر باید حاوی ارتفاع بلاک در تراکُنش پایگاه‌سکه می‌بودند.

بعد از علامت‌دهی و فعال‌سازی موفق تحت قواعد BIP-34، این ساز و کار دو بار دیگر برای فعال‌سازی دو انشعاب نرم به کار گرفته شد:

- استاندارد BIP-66 (اجباری شدن کُدگذاری DER برای امضاهای دیجیتال) با شیوه‌ی علامت‌دهی BIP-34 فعال شد: این بار از ویرایش بلاک «۳» استفاده شده و ویرایش بلاک «۲» نامعتبر شد.
- استاندارد BIP-65 (قفل زمانی CHECKLOCKTIMEVERIFY) با شیوه‌ی علامت‌دهی BIP-34 فعال شد: در این فرآیند از ویرایش بلاک «۴» استفاده شده و ویرایش بلاک «۳» نامعتبر شد.

بعد از فعال‌سازی BIP-65، ساز و کار علامت‌دهی و فعال‌سازی BIP-34 کنار گذاشته شد و جای خود را به ساز و کار علامت‌دهی BIP-9 (که در قسمت بعد توضیح می‌دهیم) داد. [برای آشنایی بیشتر با استاندارد BIP-34 می‌توانید به https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki مراجعه کنید.]

علامت‌دهی و فعال‌سازی BIP-9

ساز و کار به کار رفته در BIP-34، BIP-66 و BIP-65 در فعال‌سازی سه انشعاب نرم موفقیت‌آمیز بود. با این حال، به خاطر چند محدودیت مهم لازم بود ساز و کار بهتری جایگزین آن شود:

- به دلیل استفاده از مقادیر صحیح برای ویرایش بلاک، در هر زمان فقط یک انشعاب نرم می‌توانست فعال‌سازی شود، در نتیجه لازم بود از قبل بر سر اولویت‌بندی و ترتیب اجرای پیشنهادهای انشعاب نرم هماهنگی و توافق صورت گیرد.
- از آنجا که ویرایش بلاک فقط می‌توانست افزایش داده شود، در این ساز و کار هیچ راه ساده‌ای برای رد کردن یک تغییر و پیشنهاد دادن تغییری متفاوت وجود نداشت. اگر یک مشتری قدیمی همچنان در حال کار در شبکه بود، احتمال داشت علامت‌دهی یک تغییر جدید را با علامت‌دهی تغییری که قبلاً رد شده بود، اشتباه بگیرد.
- هر تغییری به طور برگشت‌ناپذیر ویرایش‌های بلاک موجود برای تغییرات آتی را کاهش می‌داد.

برای غلبه بر این چالش‌ها و بهبود آهنگ و سهولت پیاده‌سازی تغییرات آتی، استاندارد BIP-9 پیشنهاد شد. در BIP-9، فیلد ویرایش بلاک (به جای عدد صحیح) به عنوان یک فیلد بیتی تعبیر می‌شود. از آنجا که قبلاً اعداد صحیح ۱ تا ۴ به کار رفته‌اند، ۲۹ بیت باقی می‌ماند که می‌توانند به به عنوان یک فیلد بیتی به کار گرفته شوند. از این ۲۹ بیت می‌توان برای علامت‌دهی مستقل و همزمان ۲۹ پیشنهاد مختلف استفاده کرد.

در BIP-9، همچنین حداکثر زمان علامت‌دهی و فعال‌سازی یک پیشنهاد مشخص می‌شود. با این روش دیگر نیازی نیست معدنچیان برای همیشه علامت بدهند. اگر یک پیشنهاد در بازه‌ی زمانی TIMEOUT (که در داخل پیشنهاد تعریف شده) فعال نشود، آن پیشنهاد مردود تلقی خواهد شد. البته این پیشنهاد را می‌توان با یک بیت متفاوت مجدداً منتشر کرده و زمان رأی‌گیری آن را تمدید کرد.

علاوه بر آن، اگر بازه‌ی TIMEOUT یک پیشنهاد منقضی شده و آن پیشنهاد فعال یا مردود شود، از آن بیت می‌توان (بدون سردرگمی و ابهام) برای پیشنهاد دادن ویژگی‌های دیگر استفاده کرد. در نتیجه، امکان علامت‌دهی همزمان ۲۹ تغییر وجود دارد و بعد از سپری شدن زمان TIMEOUT، این بیت‌ها می‌توانند برای پیشنهاد دادن تغییرات جدید مجدداً به کار گرفته شوند.

> **نکته**
>
> اگر چه بیت‌های علامت‌دهی می‌توانند به دفعات مورد استفاده قرار گیرند، ولی مؤلفان BIP-9 توصیه می‌کنند تا وقتی بازه‌های رأی‌گیری همپوشانی نداشته باشند، استفاده‌ی مجدد از این بیت‌ها فقط در موارد ضروری صورت گیرد؛ به دلیل وجود باگ در نرم‌افزارهای قدیمی، استفاده‌ی مجدد از بیت‌های علامت‌دهی می‌تواند رفتارهای غیرمنتظره به دنبال داشته باشد. به بیان ساده، تا زمانی که تمامی ۲۹ بیت یک بار استفاده نشده باشند، نباید از آنها مجدداً استفاده کرد.

تغییرات پیشنهادی در یک ساختمان داده‌ی مخصوص شامل فیلدهای زیر تعریف می‌شوند:

name

توصیف کوتاهی برای تمایز بین پیشنهادهای مختلف. اکثر اوقات از استاندارد BIP توصیف‌کننده‌ی این پیشنهاد به صورت «bipN» استفاده می‌شود، که در آن N همان عدد BIP است.

bit

مقداری بین ۰ تا ۲۸؛ بیت مربوطه در فیلد ویرایش بلاک که معدنچیان برای اعلام حمایت از این پیشنهاد استفاده می‌کنند.

starttime

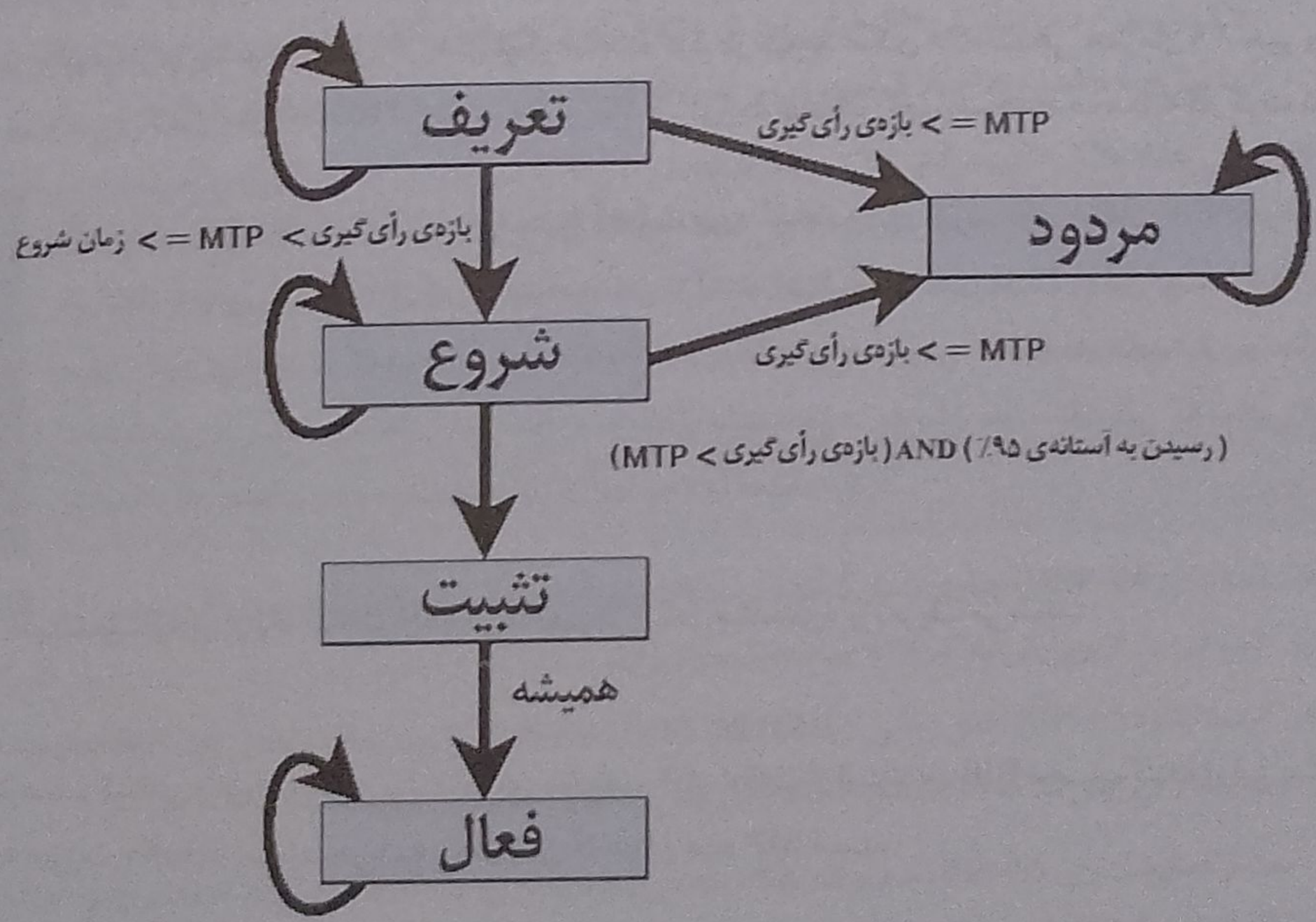
زمان شروع علامت‌دهی (بر مبنای MTP: زمان سپری‌شده‌ی میانه). از این زمان به بعد ست کردن بیت مربوطه به معنای اعلام موافقت با آن پیشنهاد است.

endtime

زمان پایان رأی‌گیری (بر مبنای MTP). اگر بعد از این زمان تعداد بیت‌های موافق به حد نصاب نرسیده باشد، آن پیشنهاد مردود تلقی خواهد شد.

در BIP-9 (بر خلاف BIP-34)، علامت‌های موافق در بازه‌هایی که مضربی از دوره‌ی بازتنظیم هدف دشواری (۲۰۱۶ بلاک) هستند، شمارش می‌شوند. اگر تعداد بلاک‌های دارای بیت موافق یک پیشنهاد از ۹۵ درصد (۱۹۱۶ بلاک از ۲۰۱۶ بلاک) بیشتر شود، آن پیشنهاد در دوره‌ی بازتنظیم هدف دشواری بعدی فعال خواهد شد. در BIP-9، هر پیشنهاد یک «نمودار حالت» دارد که مراحل مختلف و گذارهای آن پیشنهاد را نمایش می‌دهد؛ شکل ۱۰-۱۰ را ببینید.

یک پیشنهاد کار خود را از حالت «تعریف» (DEFINED) شروع می‌کند، که در آن پارامترهای پیشنهاد در نرم‌افزار بیت‌کوین شناخته (تعریف) می‌شوند. برای بلاک‌هایی که MTP آنها بزرگتر از زمان شروع رأی‌گیری باشد، این پیشنهاد وارد حالت «شروع» (STARTED) می‌شود. اگر در یک دوره‌ی بازتنظیم هدف دشواری تعداد بلاک‌هایی که موافقت خود با این پیشنهاد را اعلام می‌کنند (بیت مربوطه را ست می‌کنند)، از آستانه‌ی ۹۵٪ فراتر برود، و بازه‌ی زمانی رأی‌گیری به پایان نرسیده باشد، پیشنهاد وارد حالت «تثبیت» (LOCKED_IN) خواهد شد. با گذشت یک دوره‌ی بازتنظیم هدف دشواری، این پیشنهاد وارد حالت «فعال» (ACTIVE) می‌شود. همین که یک پیشنهاد وارد حالت «فعال» (ACTIVE) شد، برای همیشه در این حالت باقی خواهد ماند. اگر قبل از رسیدن به آستانه‌ی ۹۵٪



شکل ۱۰-۱۰ نمودار گذار حالت BIP-9.

دوره‌ی رأی‌گیری پایان یابد، پیشنهاد وارد حالت «مردود» (FAILED) می‌شود، که حاکی از پذیرفته نشدن آن است. پیشنهادهای «پذیرفته‌نشده» (REJECTED) نیز برای همیشه در این حالت باقی می‌مانند.

ساز و کار BIP-9 اولین بار برای فعال‌سازی ویژگی CHECKSEQUENCEVERIFY و استانداردهای وابسته (BIP-68، BIP-112 و BIP-113) به کار گرفته شد. این پیشنهاد که به آن نام «csv» داده شده بود، در ژوئیه ۲۰۱۶ با موفقیت فعال شد. [برای آشنایی بیشتر با این استاندارد به https://github.com/bitcoin/bips/blob/master/bip-0009.mediawiki مراجعه کنید.]

توسعه‌ی نرم‌افزاری اجماع

توسعه‌ی نرم‌افزاری اجماع ادامه دارد، و بحث درباره‌ی ساز و کارهای تغییر قواعد اجماع بسیار داغ است. بیت‌کوین، بنابر ماهیت خود، معیارهای هماهنگی و رسیدن به توافق بر سر تغییر این قواعد را بسیار سخت و بالا در نظر گرفته است. بیت‌کوین، به عنوان یک سیستم غیرمتمرکز، هیچ اراده‌ای برای تحمیل هر گونه «اقتدار» بر شرکت‌کنندگان در این شبکه ندارد. قدرت بین اجزای مختلف این شبکه، از معدنچیان گرفته تا آنهایی که برای هسته‌ی بیت‌کوین برنامه‌نویسی می‌کنند، آنهایی که برنامه‌های کیف پول می‌نویسند، صرافی‌ها، تاجران و کاربران نهایی پراکنده شده است. هیچ یک از این بخش‌ها نمی‌توانند یکجانبه تصمیم بگیرند. برای مثال، هر چند از لحاظ نظری معدنچیان می‌توانند با به دست گرفتن اکثریت (۵۱٪) شبکه قواعد اجماع را تغییر دهند، ولی همچنان مجبور هستند رضایت سایر بخش‌ها را نیز جلب کنند. اگر آنها یکجانبه عمل کنند، سایرین به سادگی از آنها پیروی نخواهند کرد، و به فعالیت اقتصادی در یک زنجیره‌ی اقلیت ادامه خواهند داد. بدون فعالیت اقتصادی (تراکُنش، تجارت، کیف‌پول، و داد و ستد ارز)، معدنچیان چیزی برای استخراج ندارند. این پراکندگی به معنای آن است که همه‌ی شرکت‌کنندگان باید با یکدیگر همکاری کنند، در غیر این صورت هیچ تغییری اتفاق نخواهد افتاد. یک تغییر فراگیر و پایدار فقط زمانی ممکن است که اکثریت قاطع اعضای شبکه روی آن اتفاق نظر داشته باشند. در نظر گرفتن آستانه‌ی ۹۵٪ برای تثبیت یک انشعاب نرم بازتابی از همین واقعیت است.

مهم است بدانید هیچ راه‌حل کامل و جامعی برای برنامه‌ریزی و رسیدن به اجماع وجود ندارد. انشعاب سخت و نرم هر دو متضمن بده-بستان هستند. برای برخی از انواع تغییر، انشعاب نرم گزینه‌ی بهتر است؛ در حالی که برای برخی دیگر انشعاب سخت می‌تواند انتخاب مناسب‌تری باشد. چیزی به نام «انتخاب احسن» وجود ندارد؛ هر دو نوع انشعاب با خطرات خاص خود همراه هستند. یکی از خصیصه‌های ثابت توسعه‌ی نرم‌افزاری اجماع دشوار بودن تغییر و لزوم مصالحه بین نیروهای درگیر در فرآیند اجماع است.

برخی این را یک نقطه ضعف سیستم‌های اجماع می‌دانند. اما با گذشت زمان شما هم با من هم‌عقیده خواهید شد که این بزرگترین نقطه‌ی قوت سیستم اجماع بیت‌کوین است.