ネットワークルーティング攻撃に対する
ブロックチェーンシミュレーション

Nov 21-22, 2019 IEICE NS研究会

中京大学工学研究科　松岡主馬

はじめに

- 2009年：Bitcoin 誕生
  - **分散型台帳システム（ブロックチェーン）**
    - 信頼できる第三者を必要としない
    - 改ざん耐性
- ブロックチェーン技術は様々な分野で応用，研究されている
  - 金融
  - サプライチェーン
  - エンターテイメント

CryptoKitties

- セキュリティに関する研究
  - 二重使用攻撃
  - Selfish Mining 攻撃など

はじめに

ブロックチェーンアプリケーションの開発
セキュリティに関する実験

- ブロックチェーンから設計
- 実験できる環境の用意

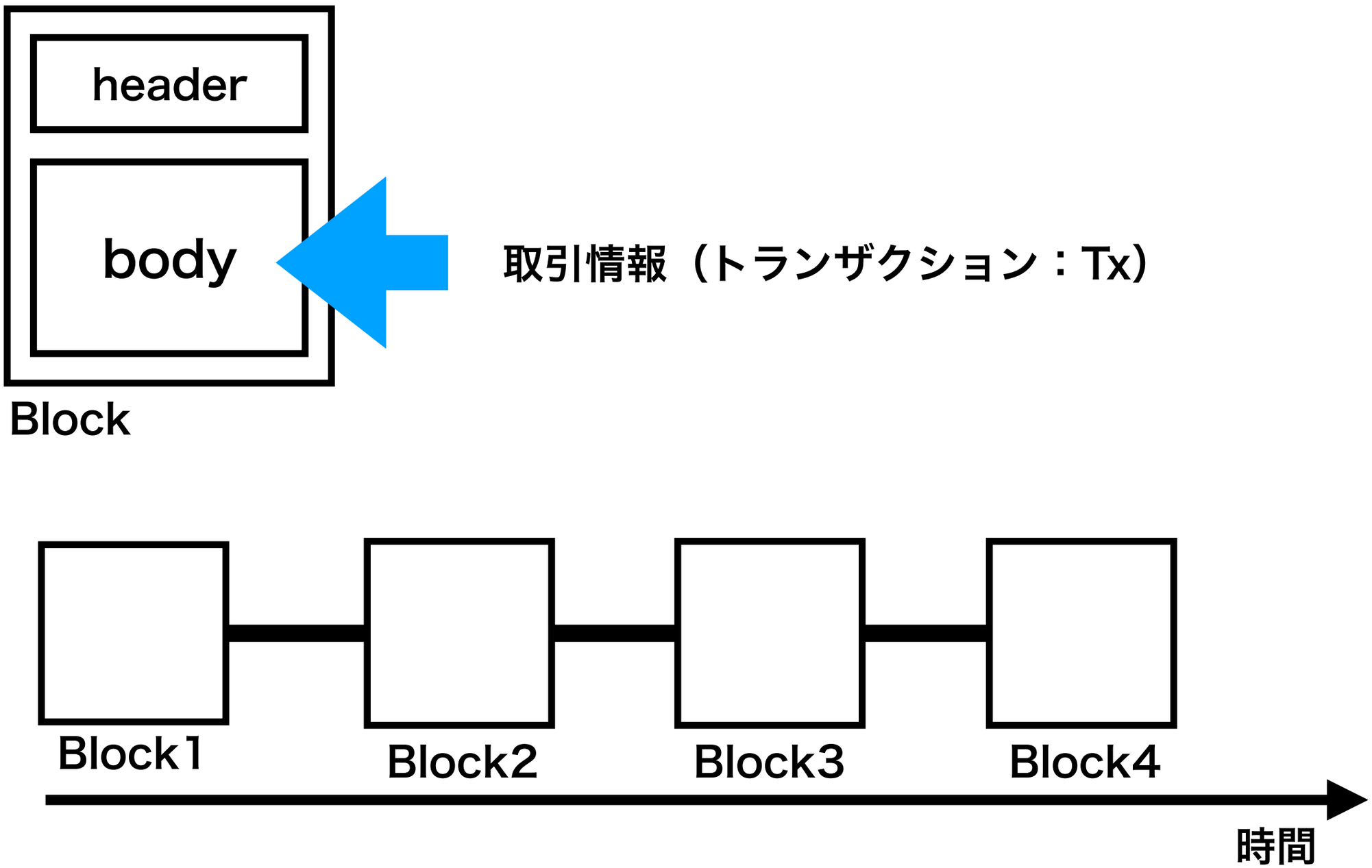
➡ ものすごい手間

アプリケーションを開発できる環境
ローカルの環境で実験できる環境

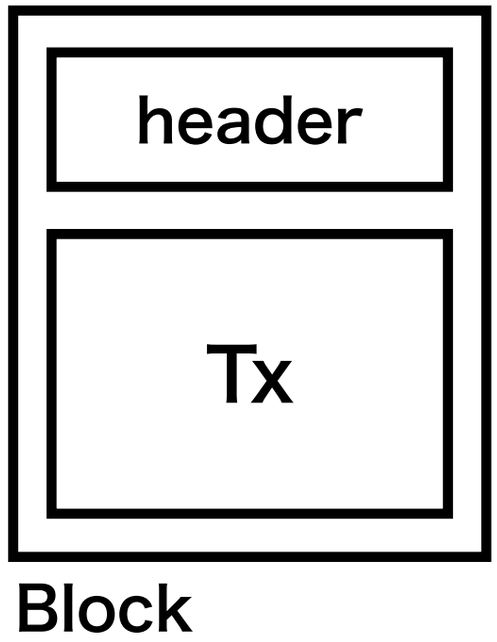
Fika

ブロックチェーンの概要

ブロックチェーン


header
body
Block
取引情報（トランザクション：Tx）
Block1
Block2
Block3
Block4
時間

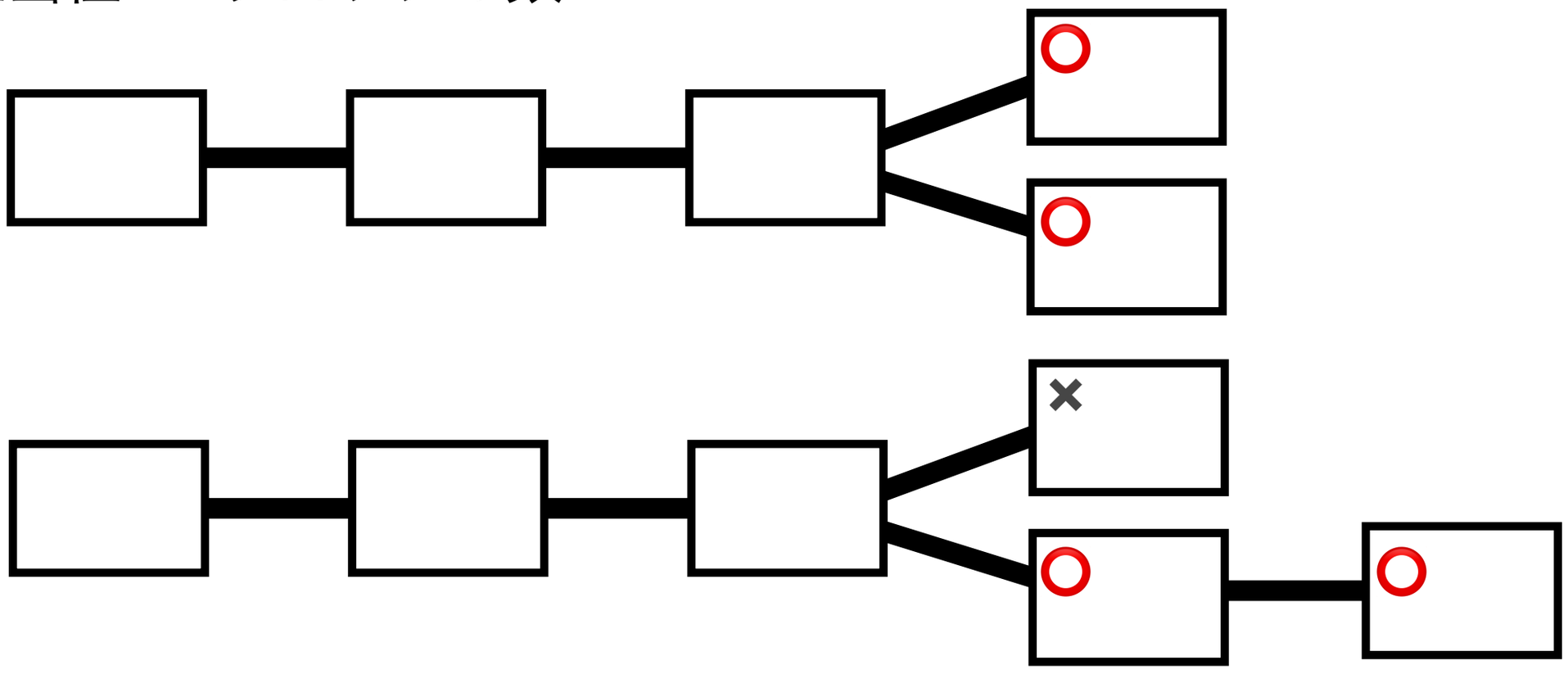
コンセンサスアルゴリズム



- ブロックの正当性を判断する
  - PoW（Proof-of-Work）
    - $hash\_function(block\_header + nonce) < target$
  - ブロックの作成
    - ナンス（nonce）を見つける計算 ➡ マイニング
  - ブロックの検証
    - ヘッダーからナンスを取り出してハッシュ値を計算

フォーク

- ブロックチェーンが複数に分岐した状態
  - ほぼ同時にブロックがマイニングされた場合
  - 仕様変更
  - 正当性 ➡ ブロックの数

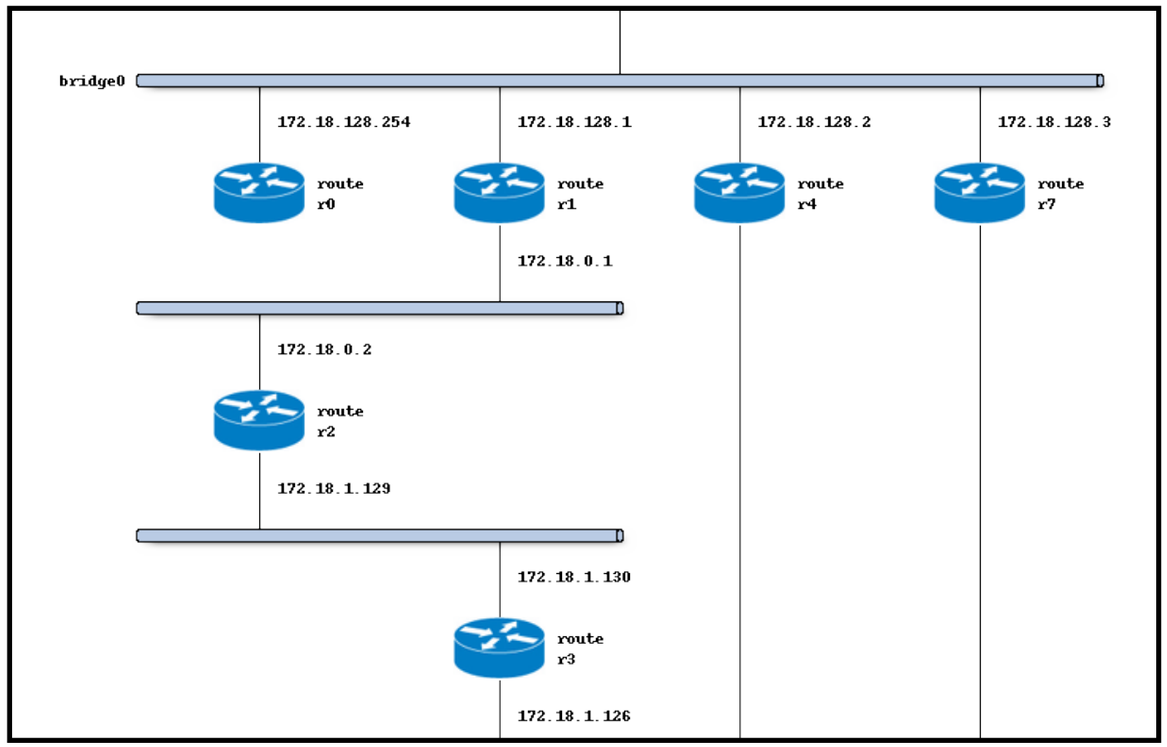


先行研究

VITOCHA

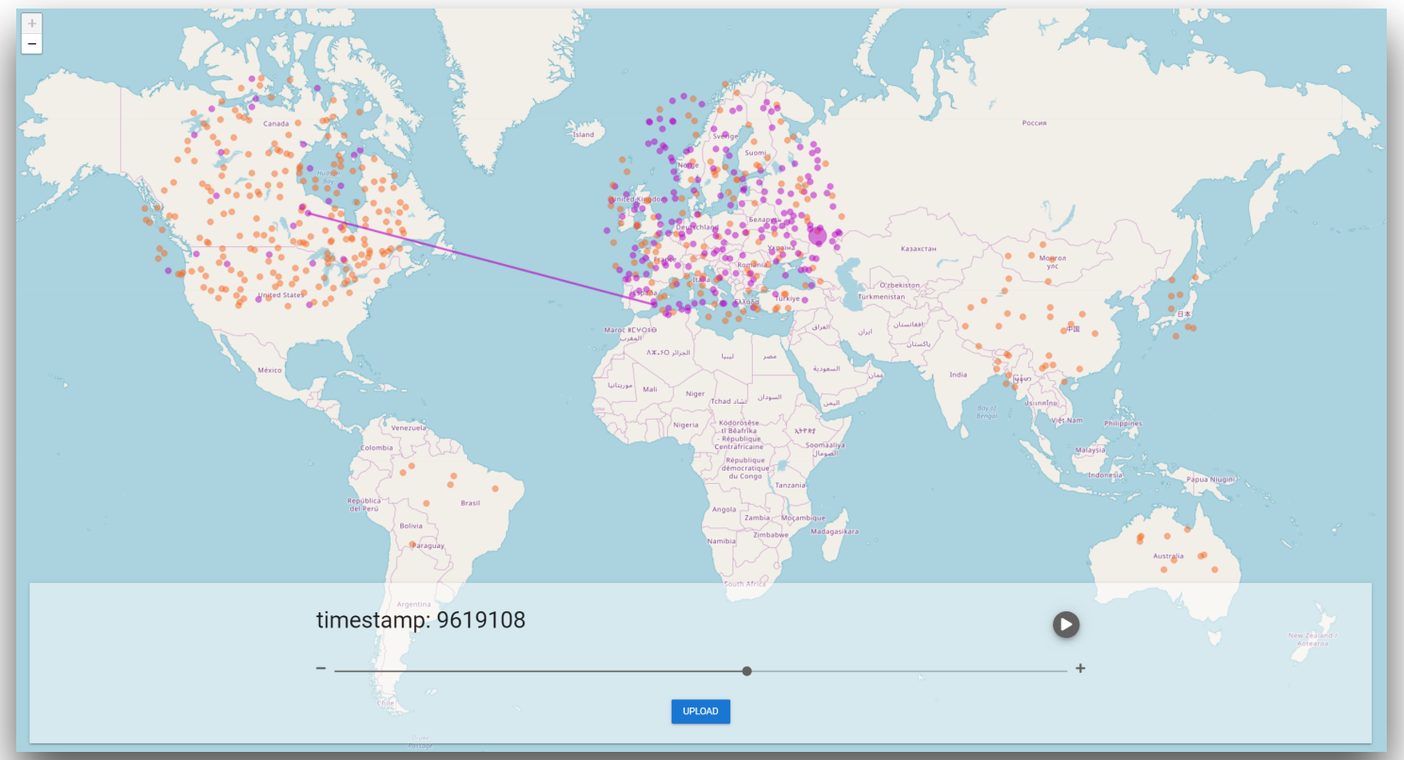
- 仮想ネットワークを1台の仮想マシンの中で構築できる Ruby ライブラリ
  - 2つの FreeBSD 機構
    - jail：ファイルシステム，プロセスの隔離
    - VIMAGE：ネットワークスタック，ルーティングテーブルの隔離

➡ Python で再実装



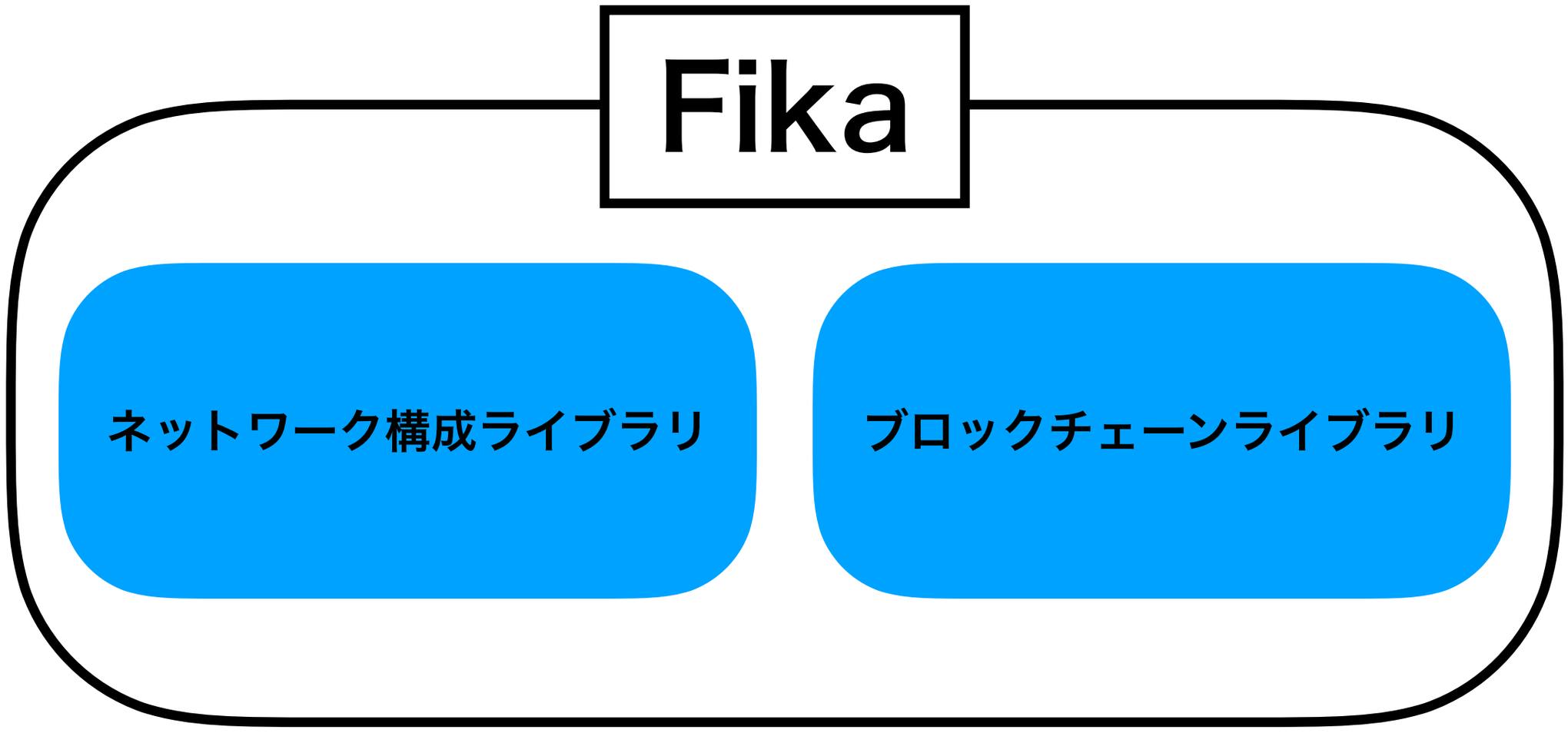
SimBlock

- ブロックチェーンネットワークを模擬するソフトウェア
  - パラメタ変更による操作
    - Bitcoin などの既存のシステム
    - 独自に考案したシステム
  - 可視化機能



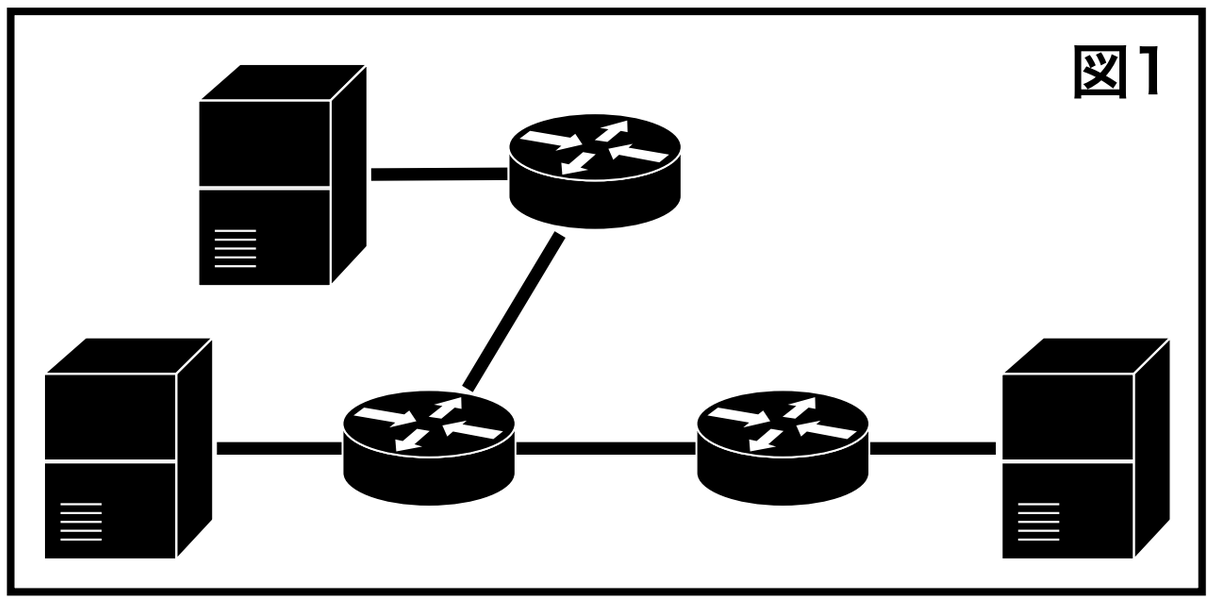
Fika の構成

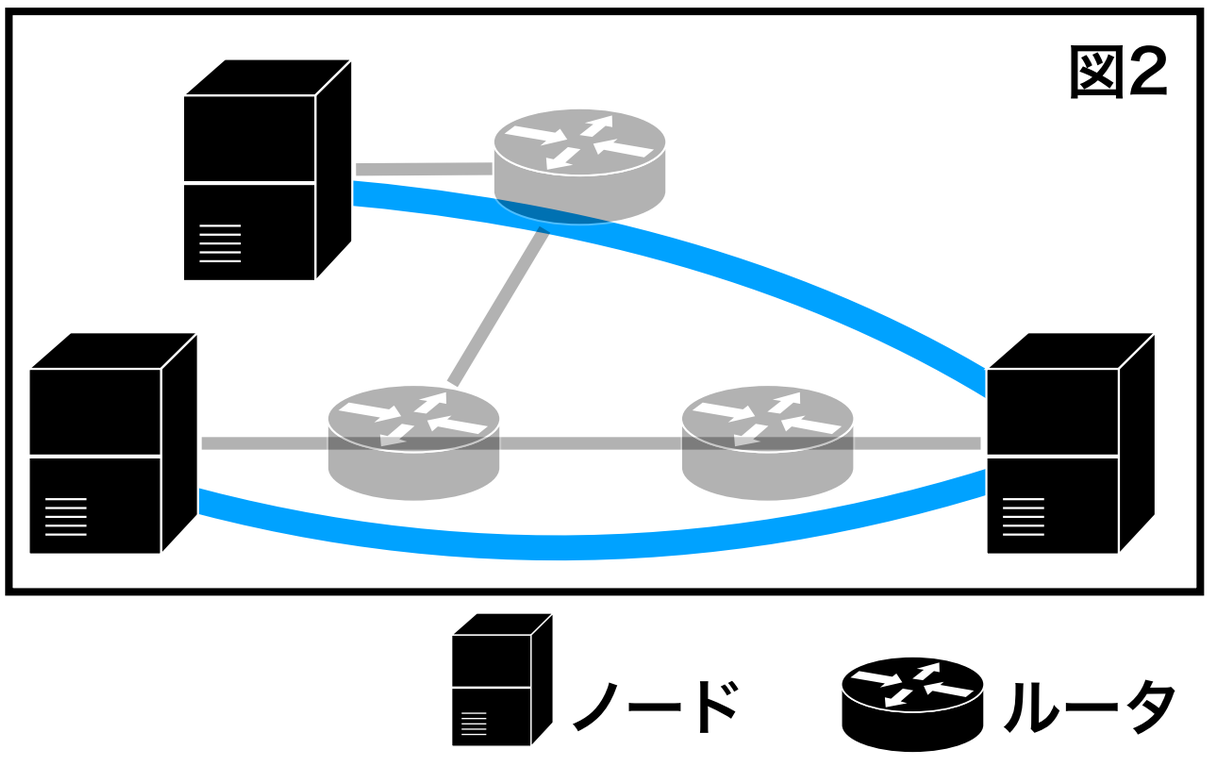
Fika の構成



ネットワーク構成ライブラリ

- VITOCHAを継承（Python）
  - 仮想ネットワークを1台のマシンの中で構築できる
  - ルータ，ノード，ブリッジの物理ネットワークの設計（図1）
  - アプリケーションにおけるノード間の論理ネットワークの設計（図2）

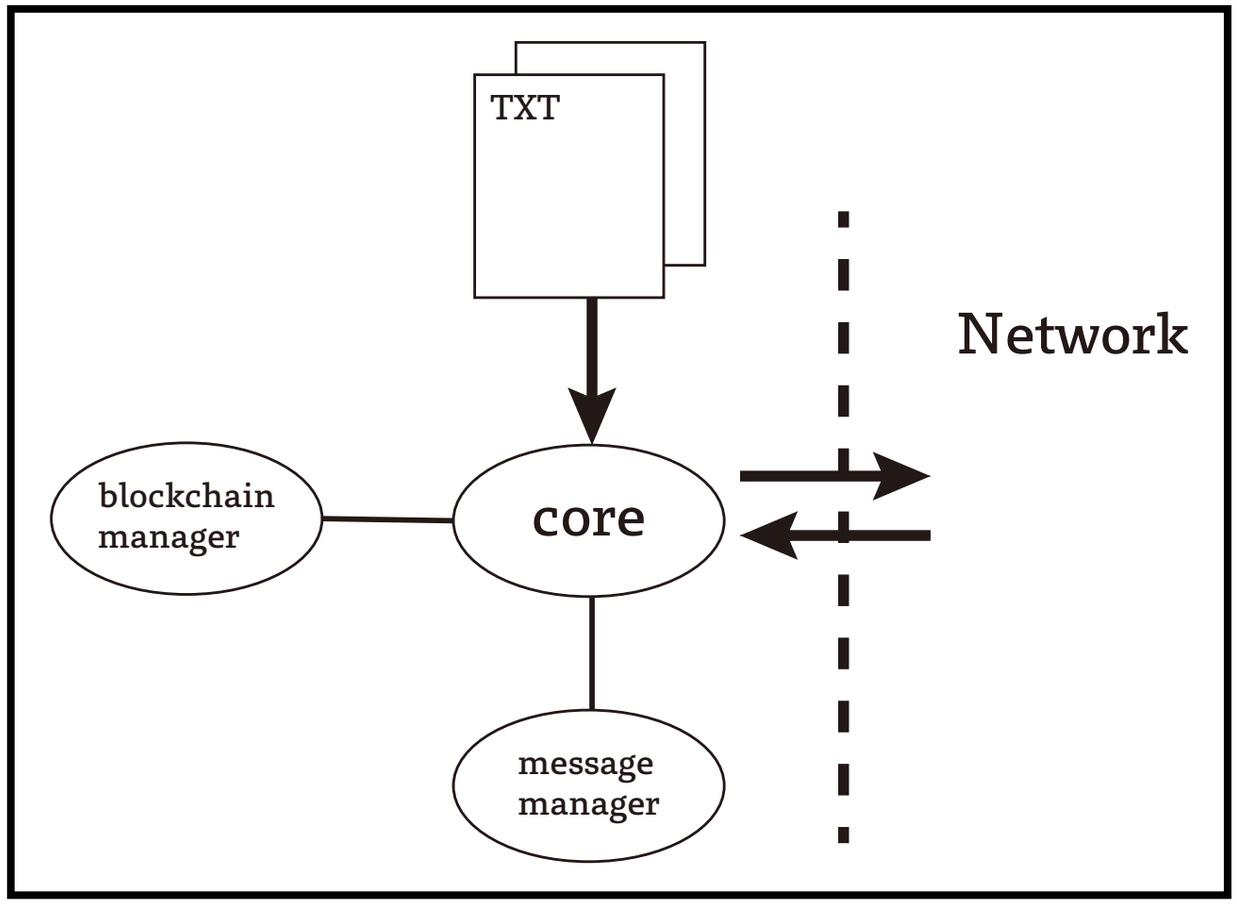




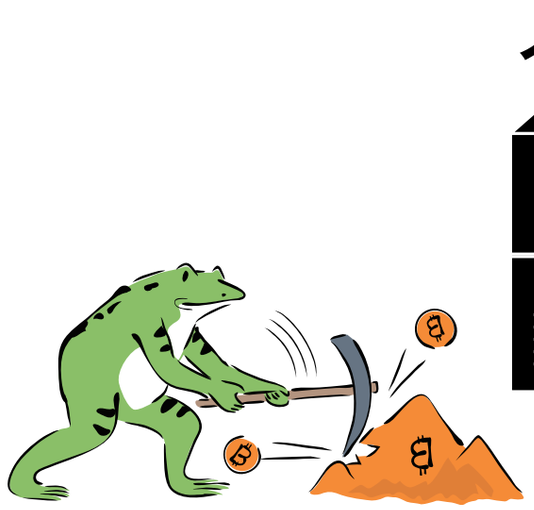
ブロックチェーンライブラリ

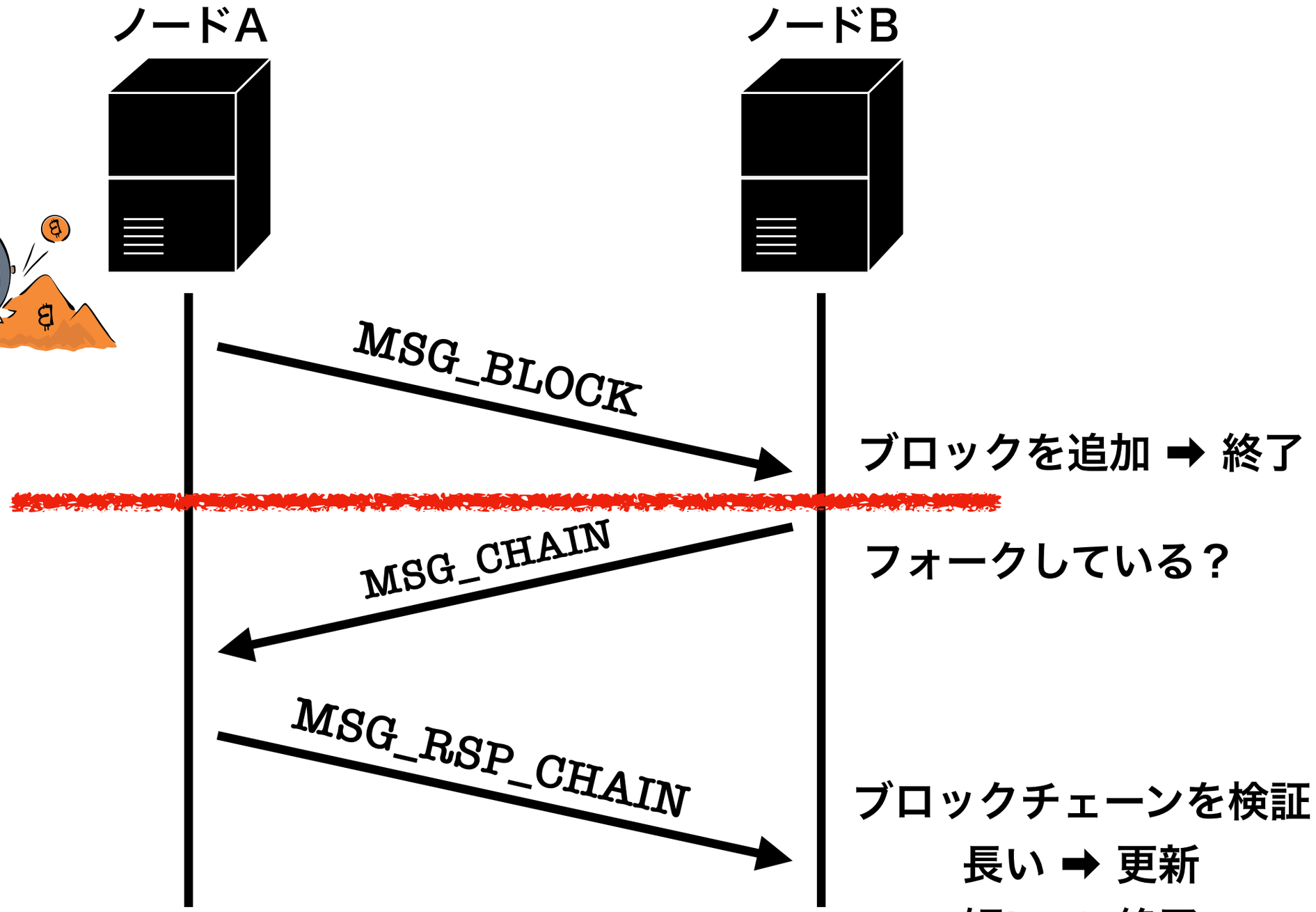
- 実験，アプリケーションを開発するために必要な要素を定義

- `message manager`
  - メッセージの解析
  - メッセージの作成
- `blockchain manager`
  - ブロックのマイニング
  - ブロック（チェーン）の検証
- `core`
  - プログラムの中枢



ブロックチェーンライブラリ



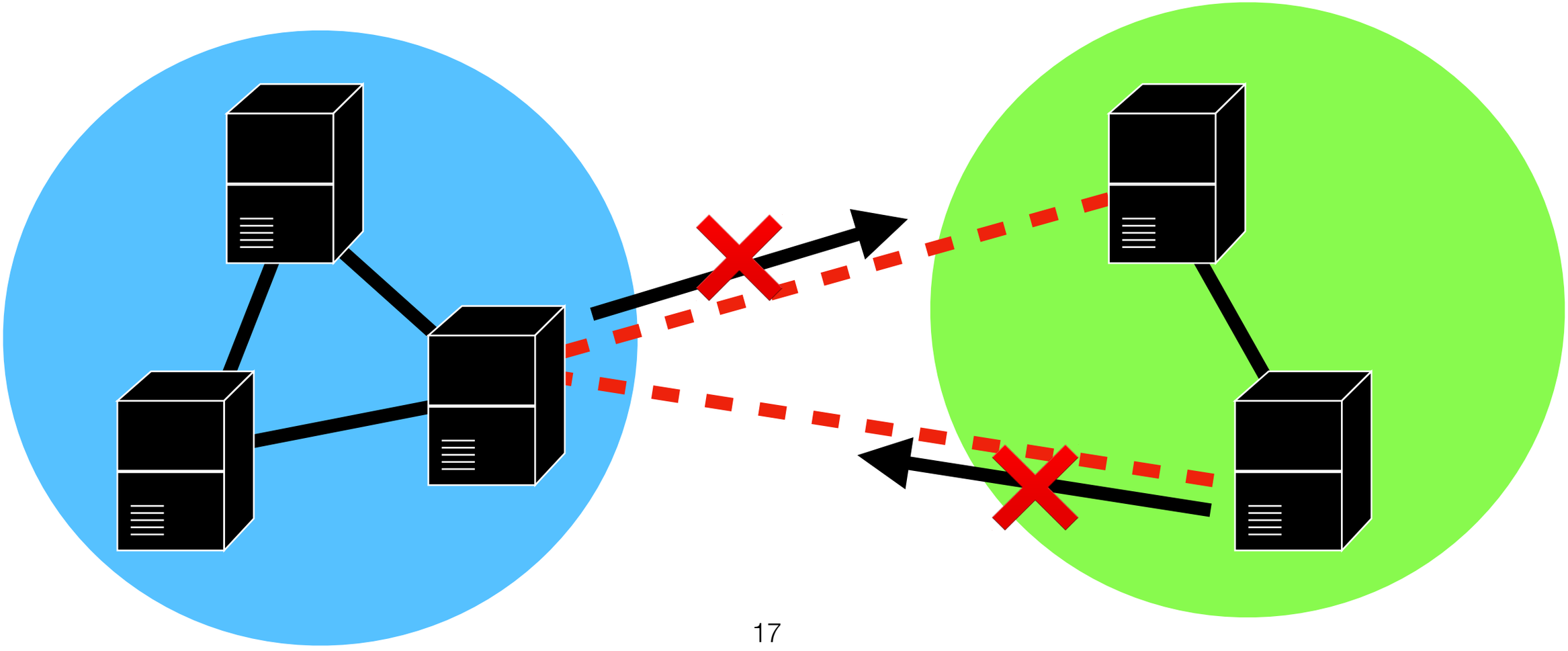


Fika の利用例

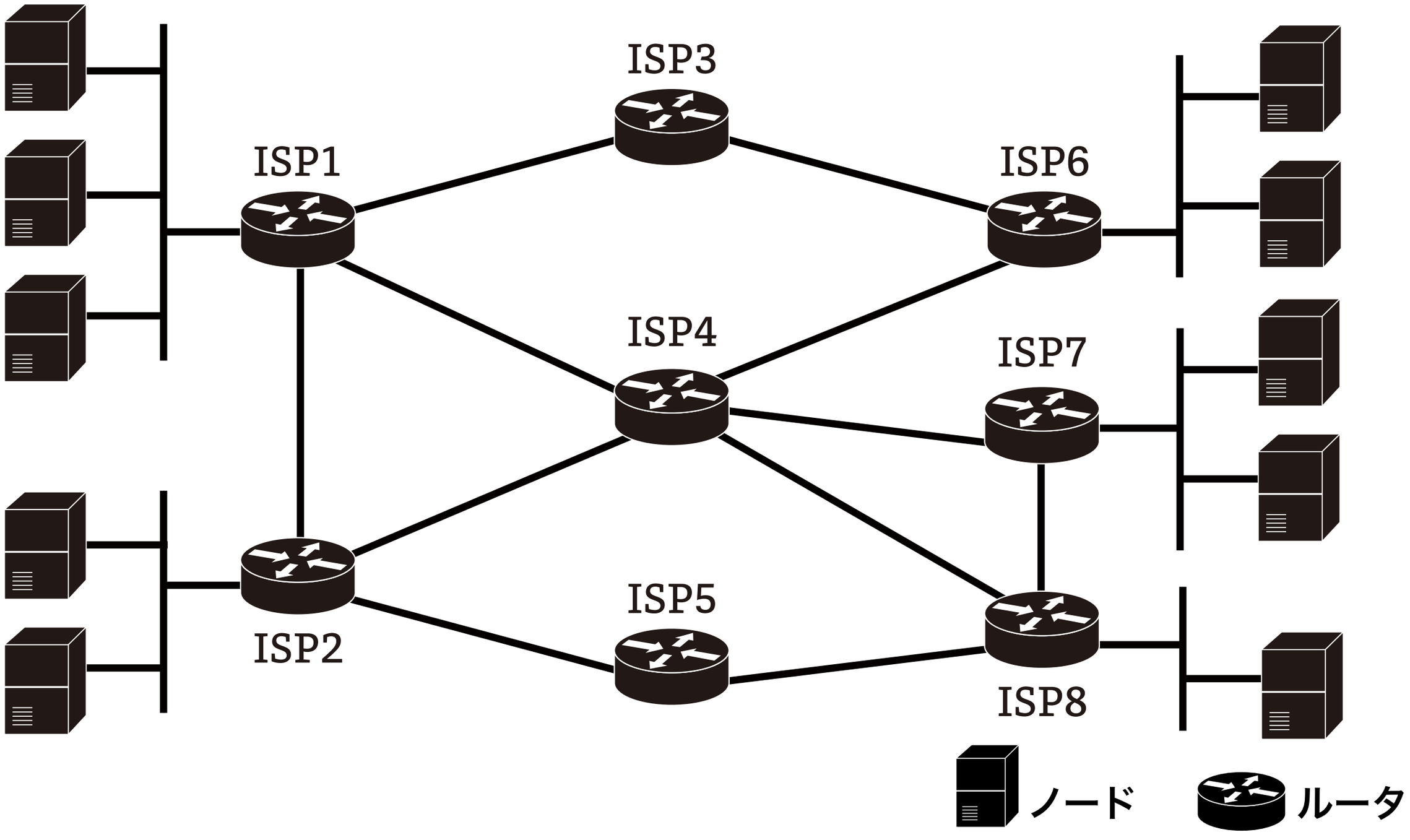
- Fika を利用して一攻撃形態のシミュレーションを行った
  - 仮想通貨システムに対するネットワークルーティング攻撃
    - 2017年，Maria Apostolaki らが提案（Partition攻撃）

Partition 攻撃

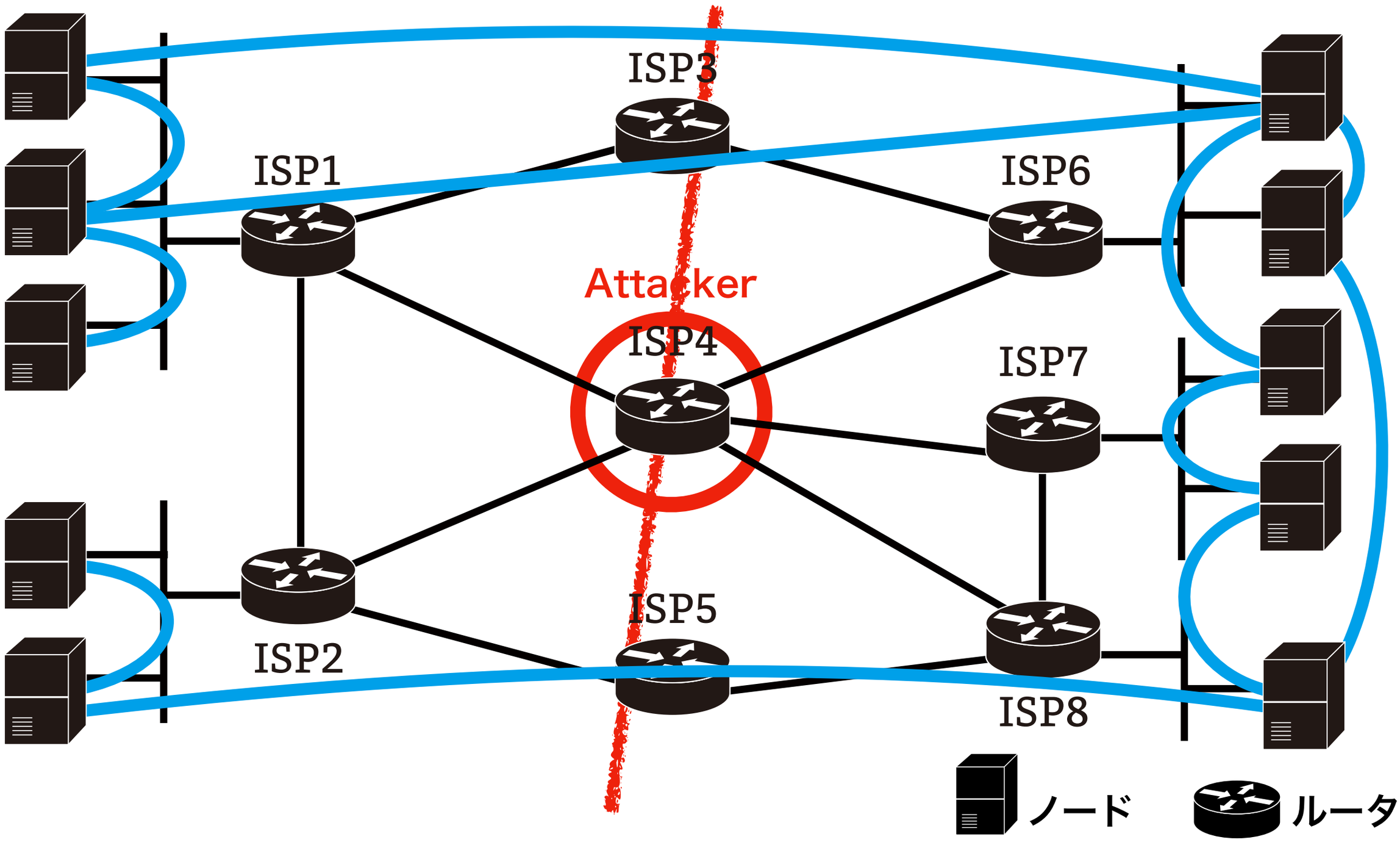
- BGP 経路ハイジャックによってブロックチェーンネットワークを相互に到達できない複数のネットワークに分割する
  - 攻撃者は意図的にフォークの引き起こすことができる



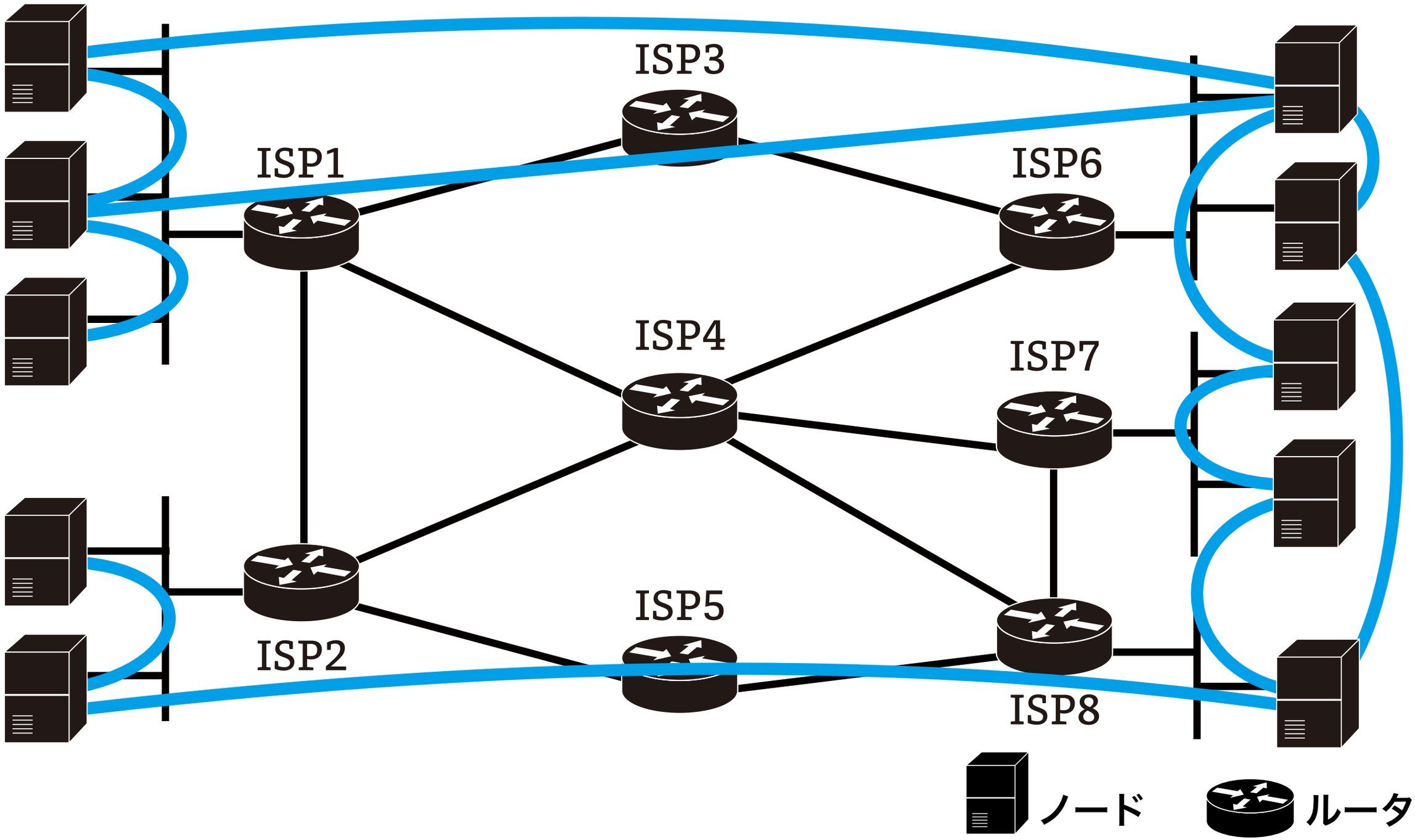
Partition 攻撃



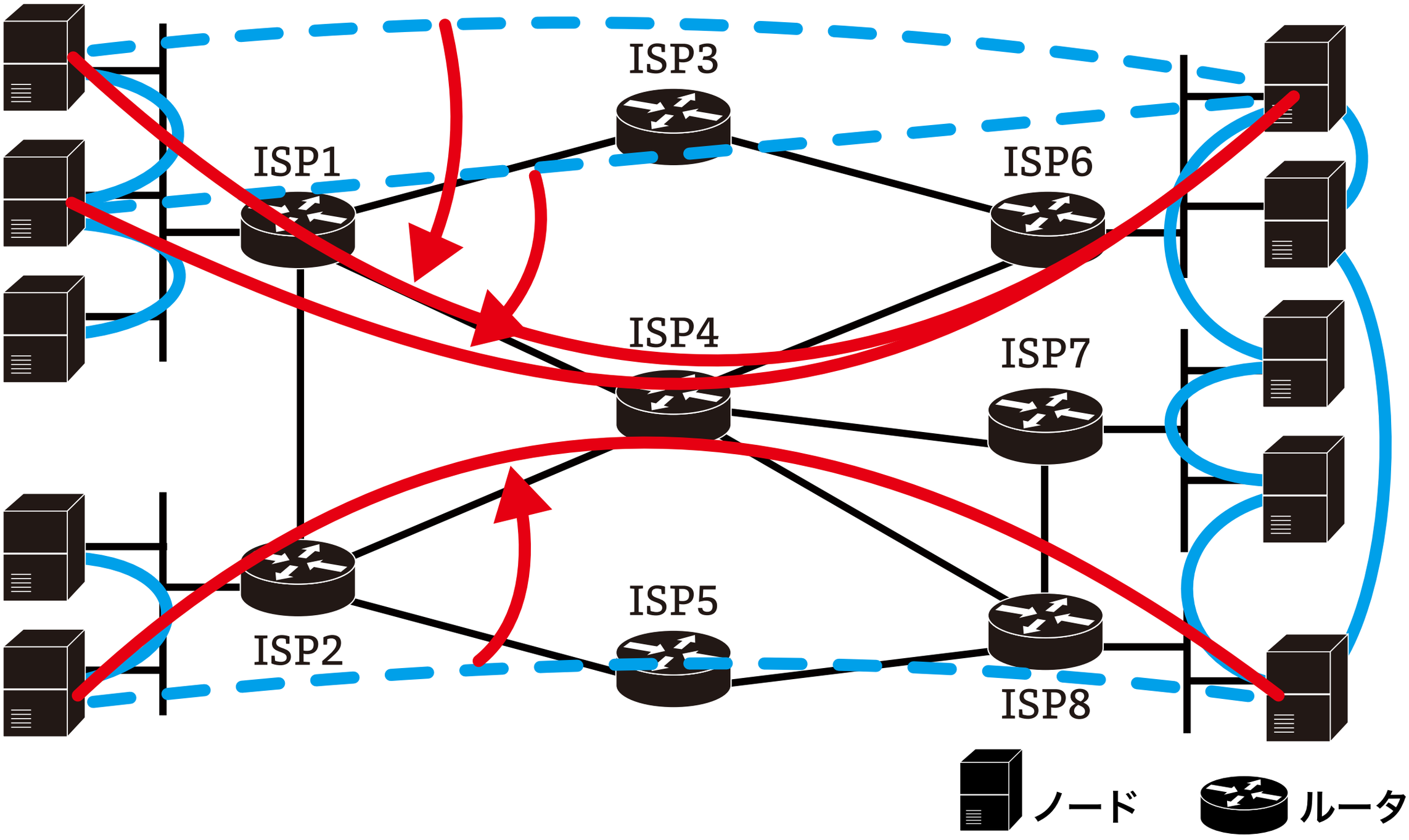
Partition 攻撃



Partition 攻撃



Partition 攻撃



Partition 攻撃

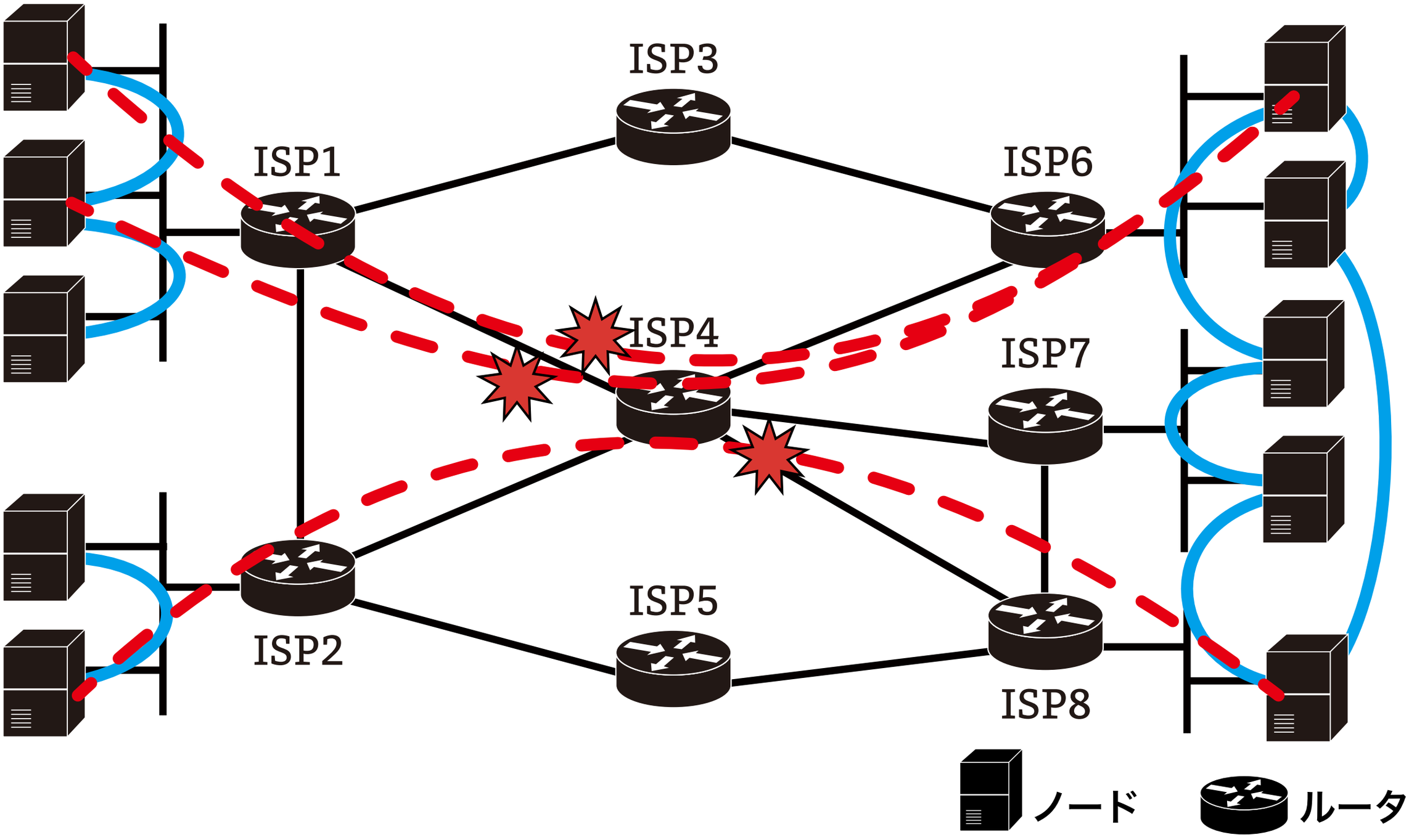
攻撃前のルーティングテーブル

```
C>* 192.168.1.0/24 is directly connected, epair3a
B>* 192.168.2.0/24 [20/0] via 172.1.2.0, epair4a, 00:06:52
B>* 192.168.3.0/24 [20/0] via 172.3.2.0, epair6a, 00:06:52
B>* 192.168.6.0/24 [20/0] via 172.3.2.0, epair6a, 00:06:49
B>* 192.168.7.0/24 [20/0] via 172.2.2.0, epair5a, 00:06:48
B>* 192.168.8.0/24 [20/0] via 172.2.2.0, epair5a, 00:06:48
r1#
```

攻撃中のルーティングテーブル

```
C>* 192.168.1.0/24 is directly connected, epair3a
B>* 192.168.1.0/25 [20/0] via 172.2.2.0, epair5a, 00:00:50
B>* 192.168.2.0/24 [20/0] via 172.1.2.0, epair4a, 00:10:46
B>* 192.168.2.0/25 [20/0] via 172.2.2.0, epair5a, 00:00:44
B>* 192.168.3.0/24 [20/0] via 172.3.2.0, epair6a, 00:10:46
B>* 192.168.6.0/24 [20/0] via 172.3.2.0, epair6a, 00:10:43
B>* 192.168.7.0/24 [20/0] via 172.2.2.0, epair5a, 00:10:42
B>* 192.168.8.0/24 [20/0] via 172.2.2.0, epair5a, 00:10:42
r1#
```

Partition 攻撃



Partition 攻撃


```
...
{
    "height": 8,
    "nonce": "37302",
    "previous_hash":"8e3d19e173913eade156266e64385
    b8228096b62d0b5d4a869ce4cc4dd1ce2b7",
    "timestamp": 1570302820.5800905
},
{
    "height": 9,
    "nonce": "121736",
    "previous_hash":"2bb0f53cafdfc405a42ea214a3fd4
    3efcc9f1660c1a386a0650804fe72ff1420",
    "timestamp": 1570302822.3661394
},
{
    "height": 10,
    "nonce": "18165",
    "previous_hash":"2405a2f73b2c2be2fc789466010ea
    4f92b03f90177700b32df6925f7a1787173",
    "timestamp": 1570302826.729632
},
{
    "height": 11,
    "nonce": "173814",
    "previous_hash":"5fe33d17d1bb7b6d48ee74477c15d
    4ac1052e5236d6fe46d2cabadab20c12621",
    "timestamp": 1570302830.773725
}
```


ノード A のブロックチェーン


```
...
{
    "height": 8,
    "nonce": "37302",
    "previous_hash":"8e3d19e173913eade156266e64385
    b8228096b62d0b5d4a869ce4cc4dd1ce2b7",
    "timestamp": 1570302820.5800905
},
{
    "height": 9,
    "nonce": "121736",
    "previous_hash":"2bb0f53cafdfc405a42ea214a3fd4
    3efcc9f1660c1a386a0650804fe72ff1420",
    "timestamp": 1570302822.3661394
},
{
    "height": 10,
    "nonce": "65708",
    "previous_hash":"2405a2f73b2c2be2fc789466010ea
    4f92b03f90177700b32df6925f7a1787173",
    "timestamp": 1570302831.4974113
},
{
    "height": 11,
    "nonce": "12189",
    "previous_hash":"25de6d98f446d3c1030147230c0ac
    26d72674e486198fccec50a26bc4e853950",
    "timestamp": 1570302836.7626328
}
]
```

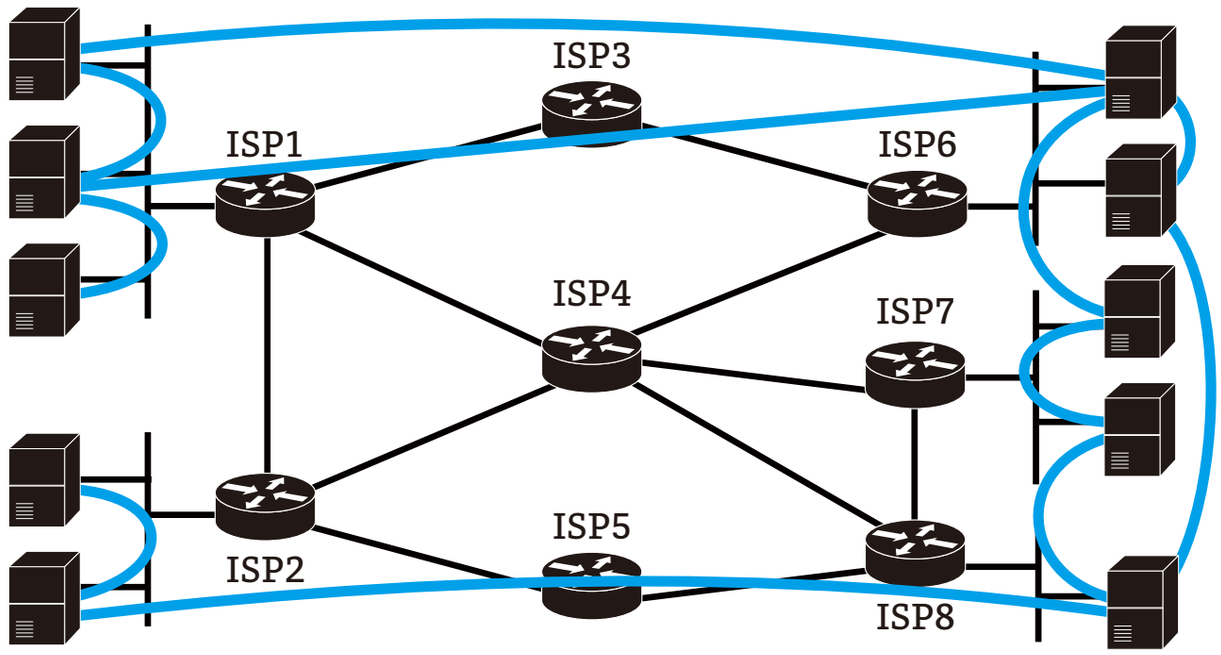

ノード B のブロックチェーン

評価

- SimBlock に比べてシミュレーションに手間がかかる
  - ネットワーク設計が必要
- メモリ消費
  - 約120MB（今回のシミュレーションの場合）
  - 1ブロック：約24KB/node
    - ブロックチェーンのデータ
    - ログ出力
  - メモリの消費効率が悪い

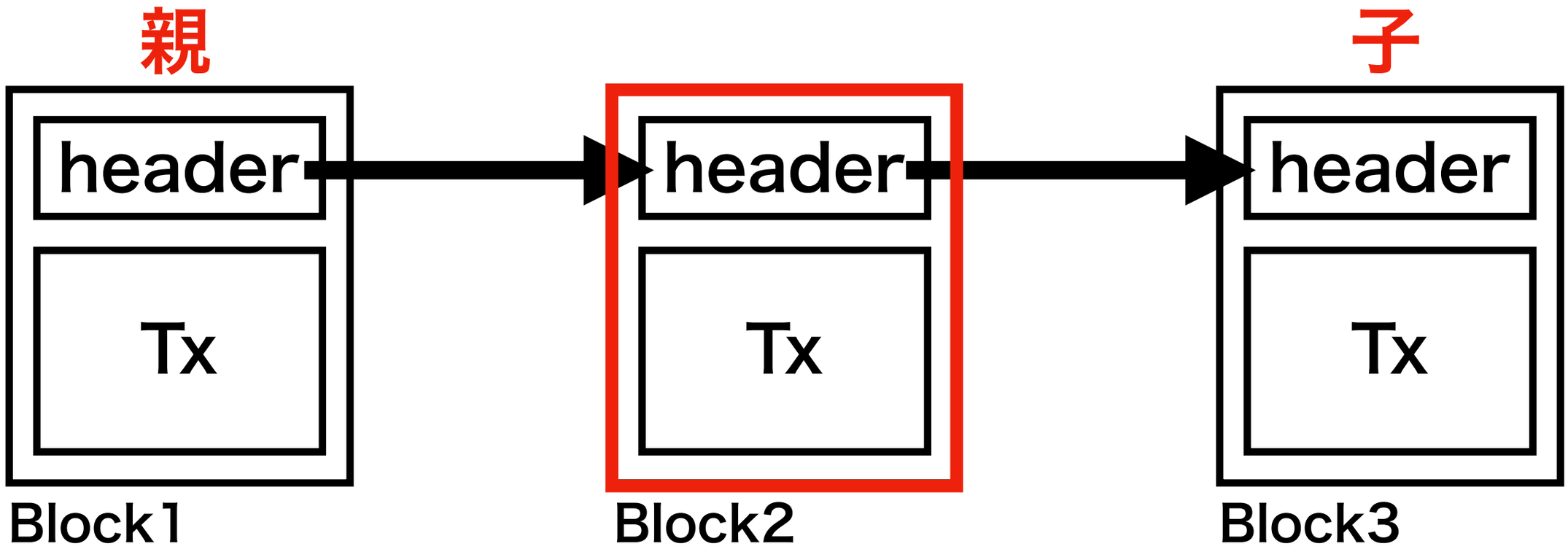
課題

- ノード数
  - 1ノード/ jail
  - 大規模ネットワークの構築が困難
- フレームワーク要素の追加
  - プログラム間の依存が強い
- 自動化
  - ノード間の接続管理
  - ルーティング設定が手作業



Isolated mining power	*min. # pfxes to hijack*	*median # pfxes to hijack*	*# feasible partitions*
8%	32	70	14
30%	83	83	1
40%	37	80	8
47%	39	39	1

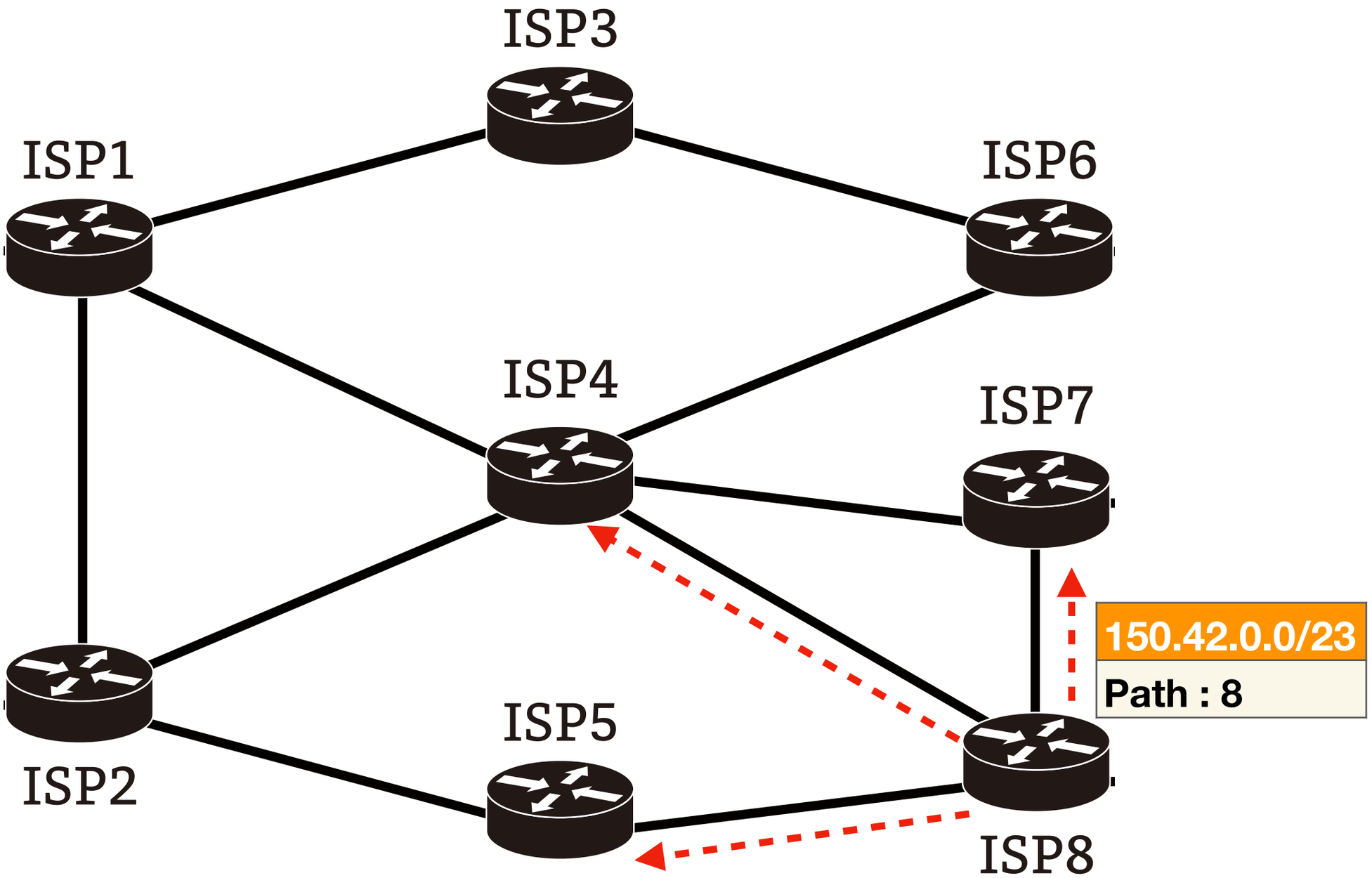
ブロックチェーン



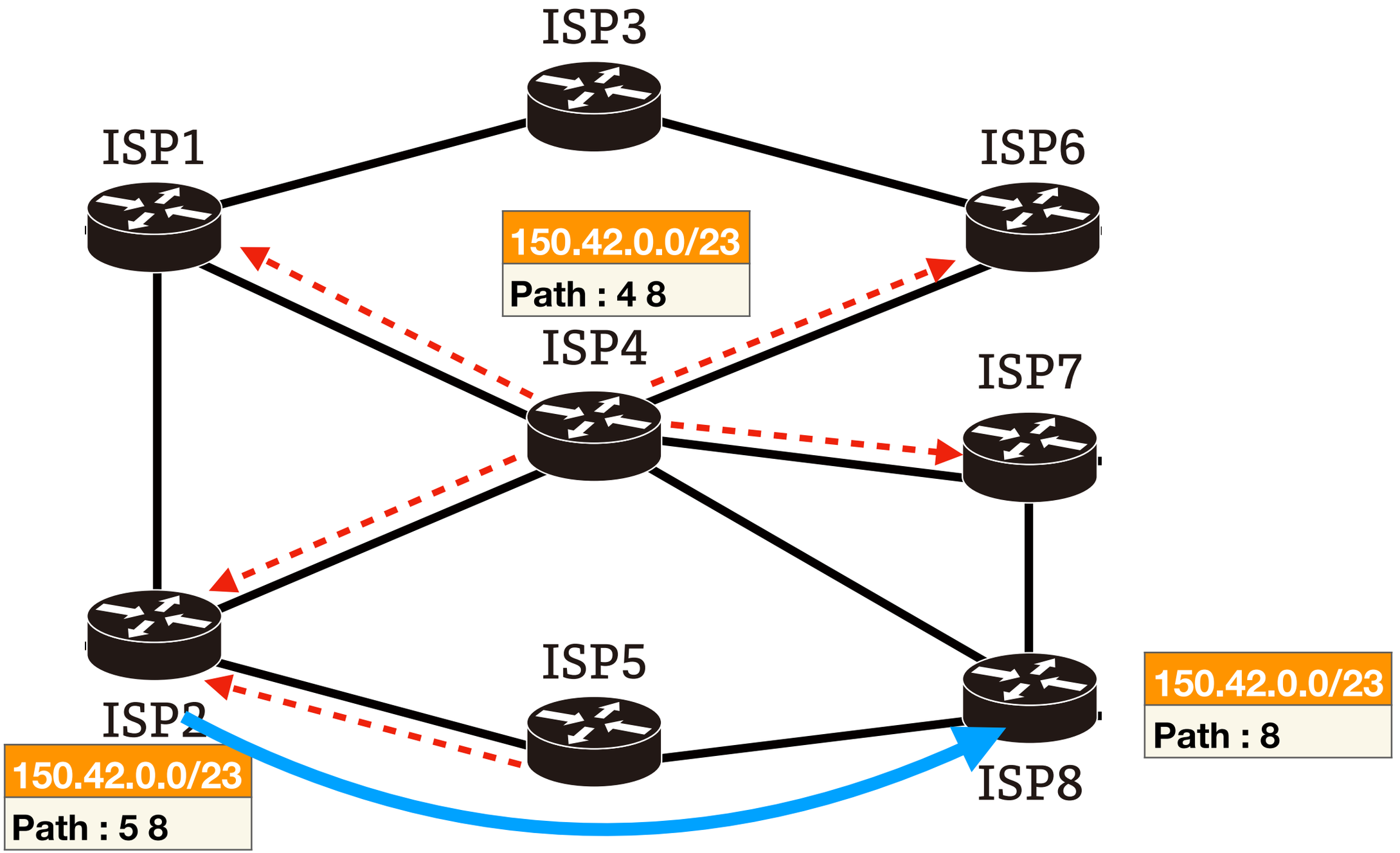
評価

- Fika の特徴
  - 機器の物理ネットワークを設計できる
  - アプリケーションにおけるノード間の論理ネットワークが設計できる
  - ネットワークにパケットを流すことができる
  - ブロックチェーンライブラリによってユーザ独自のアプリケーションを開発できる

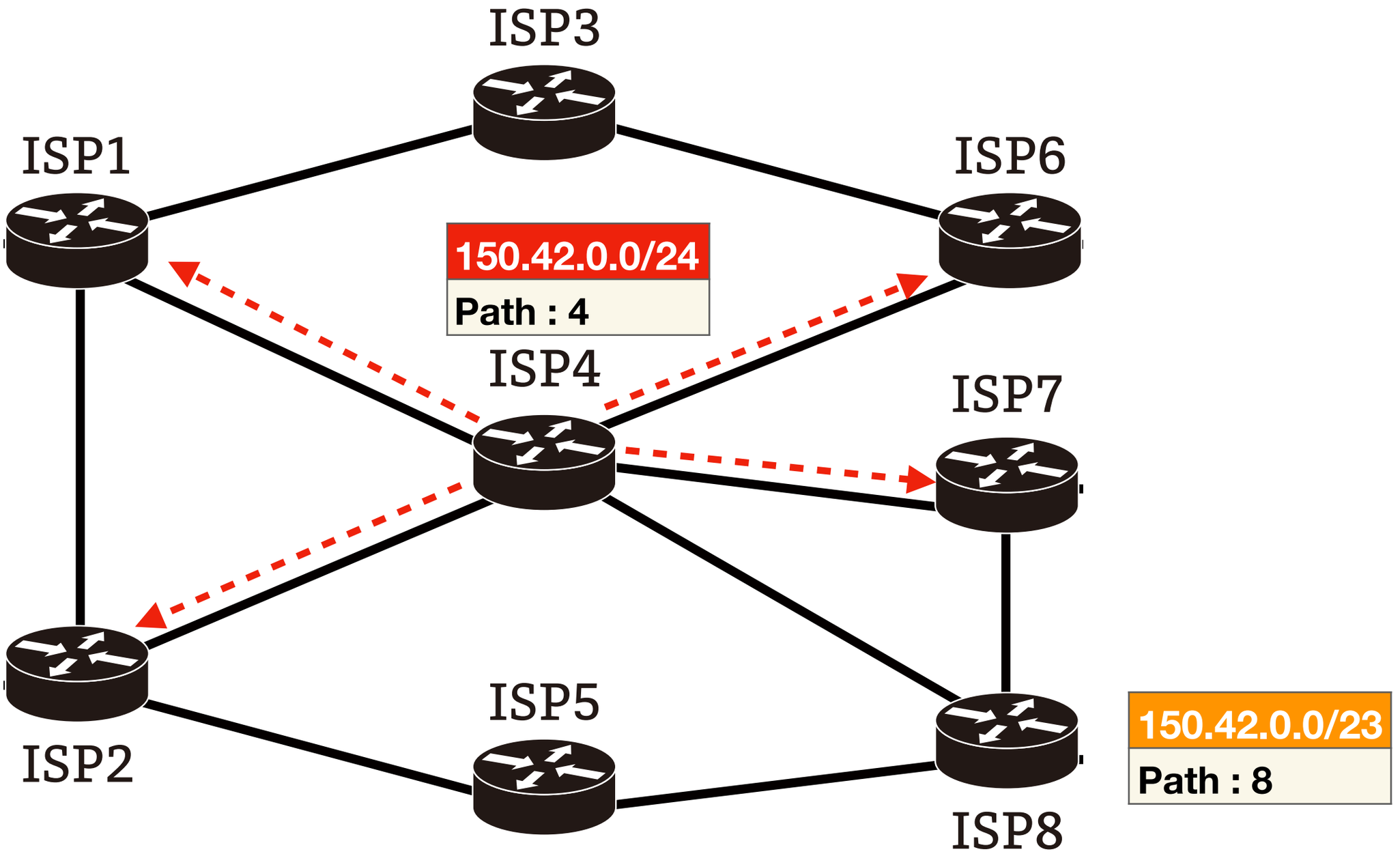
BGP経路ハイジャック



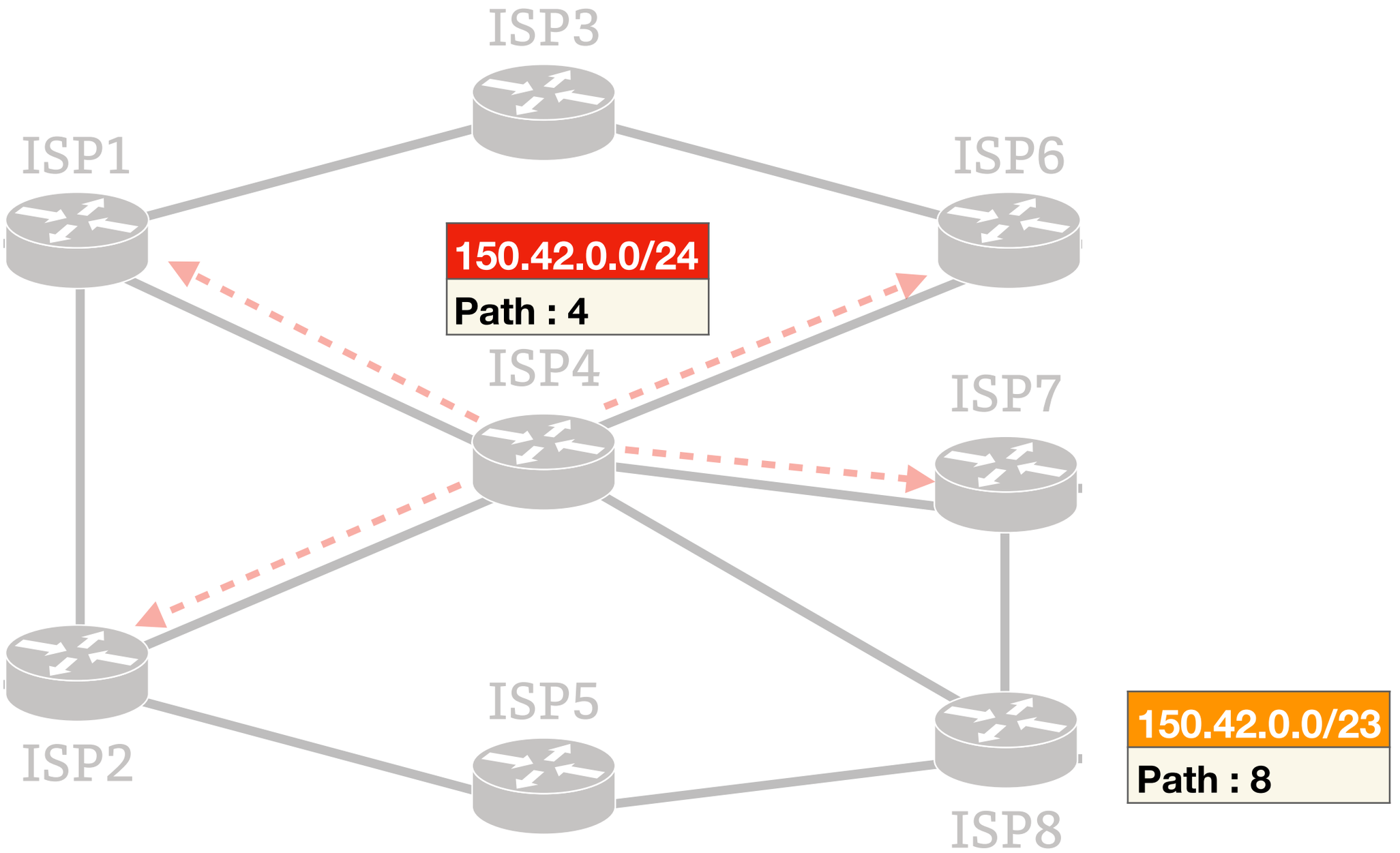
BGP経路ハイジャック



BGP経路ハイジャック



BGP経路ハイジャック



BGP経路ハイジャック