



Collusion-resilient credit-based reputation for peer-to-peer content distribution

Nguyen Tran, Jinyang Li, Lakshminarayanan
Subramanian

New York University
NetEcon'10

Incentive in P2P CDNs

A solved problem?

- Yes
 - BitTorrent tit-for-tat provides incentives for nodes to upload during download
- No
 - No incentives for nodes to act as seeders (seeder promotion problem)

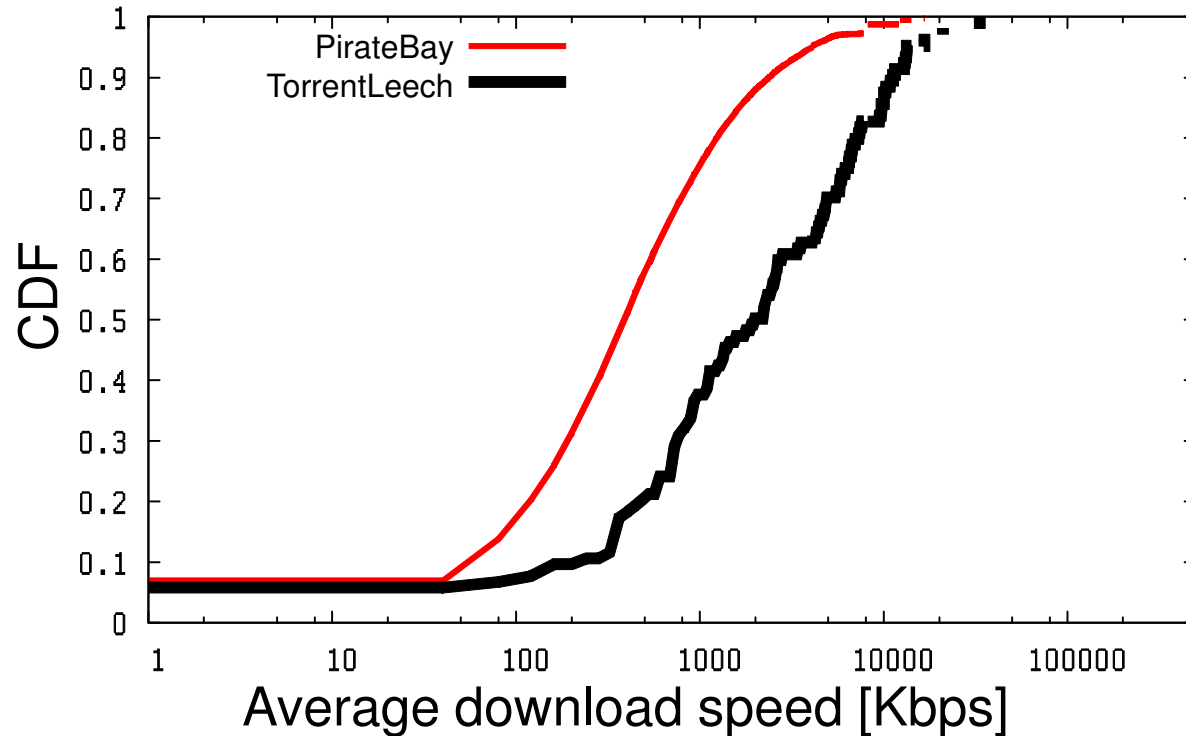
Incentive in P2P CDNs

A solved problem?

- Yes
 - BitTorrent tit-for-tat provides incentives for nodes to upload during download
- No
 - No incentives for nodes to act as seeders (seeder promotion problem)



Private vs public BitTorrent communities



More seeders → better performance

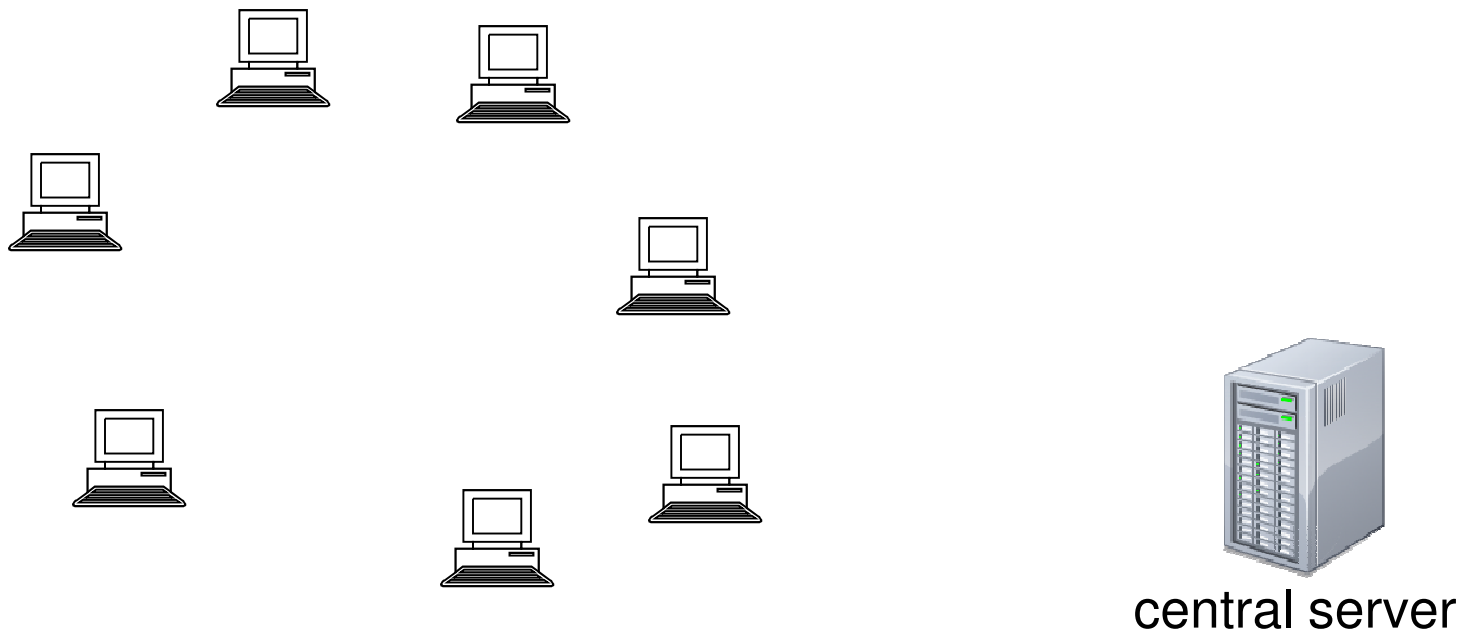
Robust reputations → seeder promotion

- Private BitTorrent
 - Nodes report their contribution → vulnerable
- Graph-based reputation (Page-rank, max-flow)
 - not capture node contribution
 - vulnerable to collusion

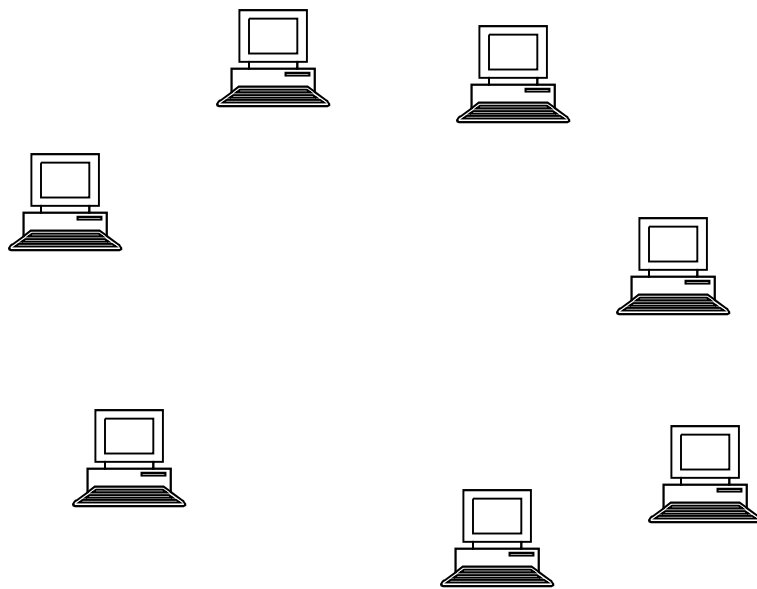
Credo: a credit-based reputation mechanism

- capture node contribution correctly
- resilient to attacks (Sybil attack and collusion)

Credo's system architecture



Credo's system architecture

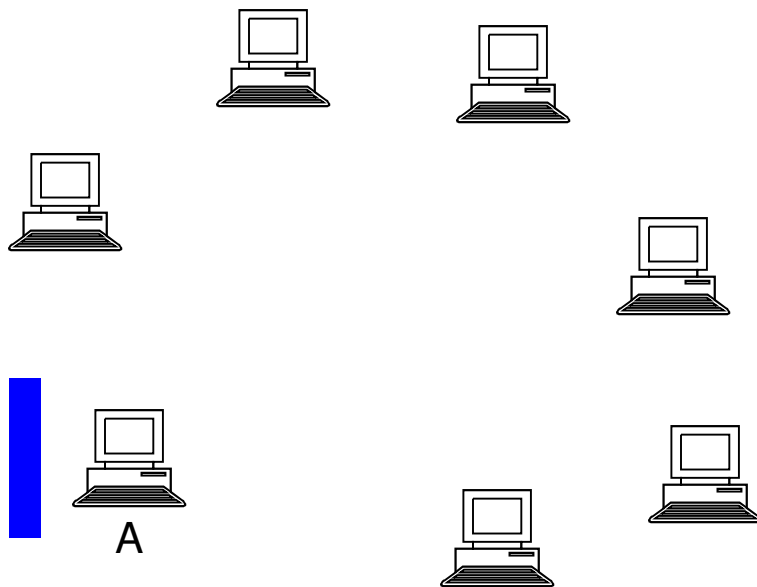


- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils

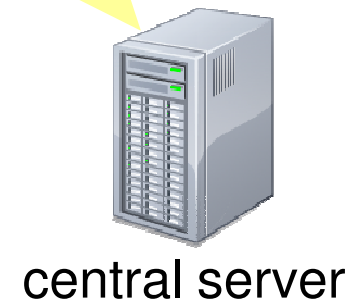


central server

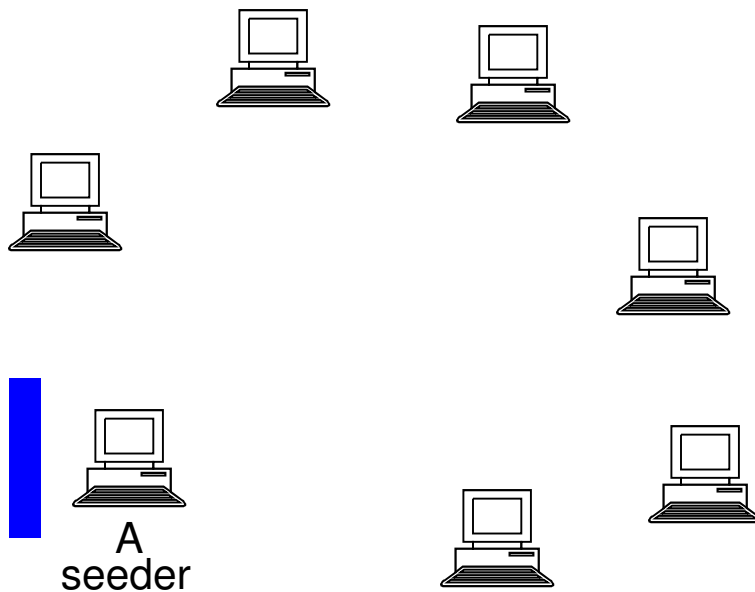
Credo's system architecture



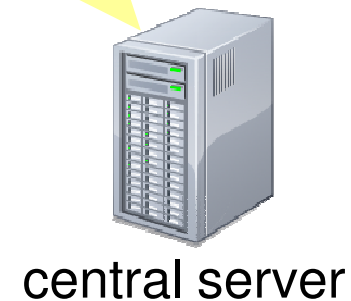
- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils



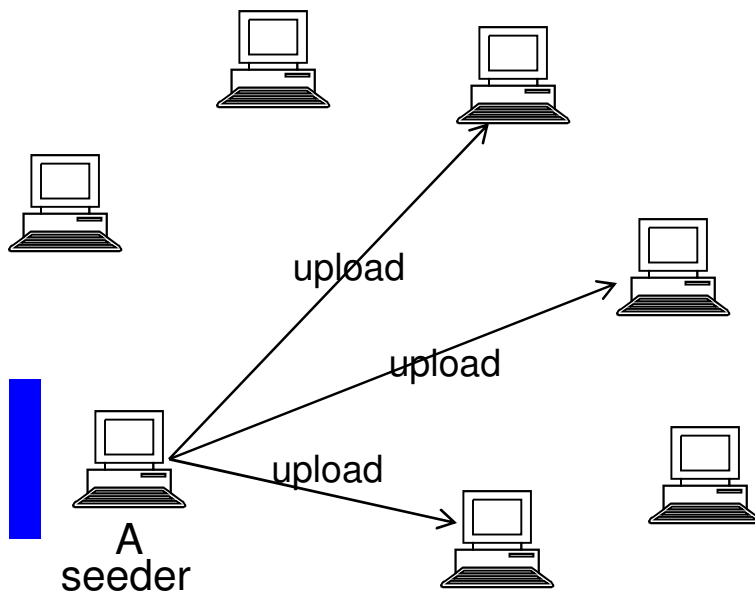
Credo's system architecture



- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils



Credo's system architecture

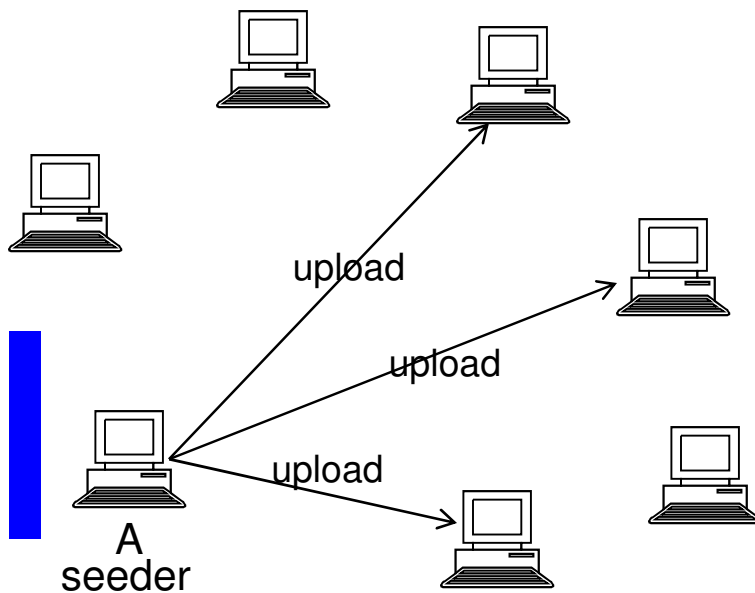


- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils



central server

Credo's system architecture

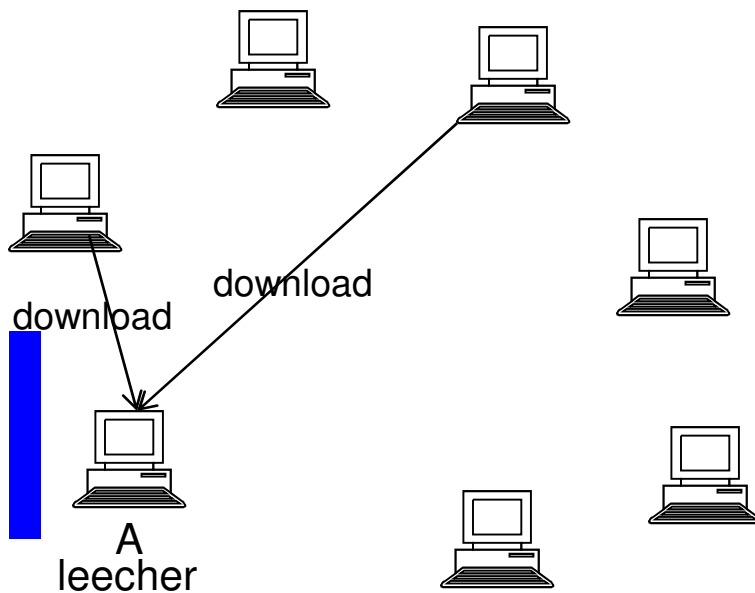


- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils



central server

Credo's system architecture

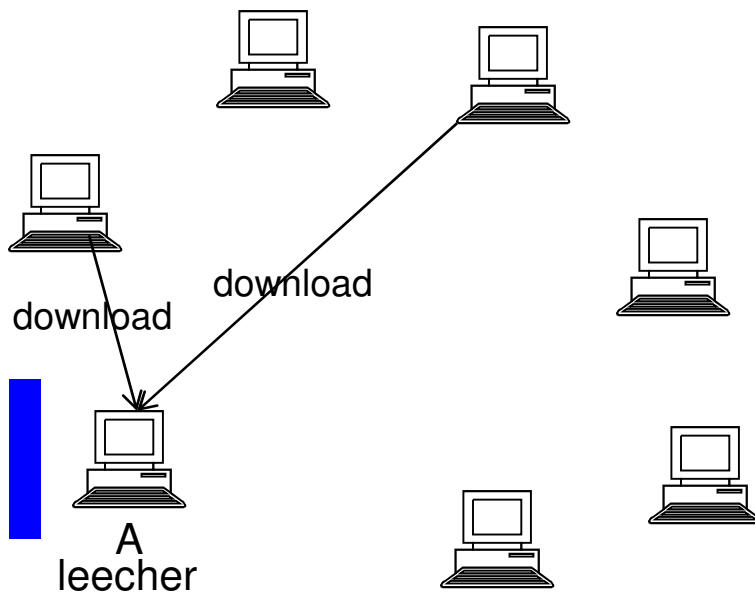


- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils

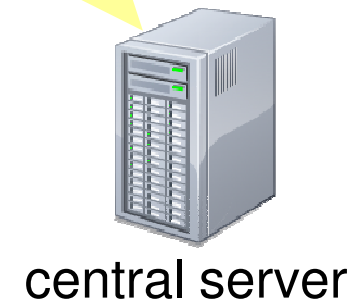


central server

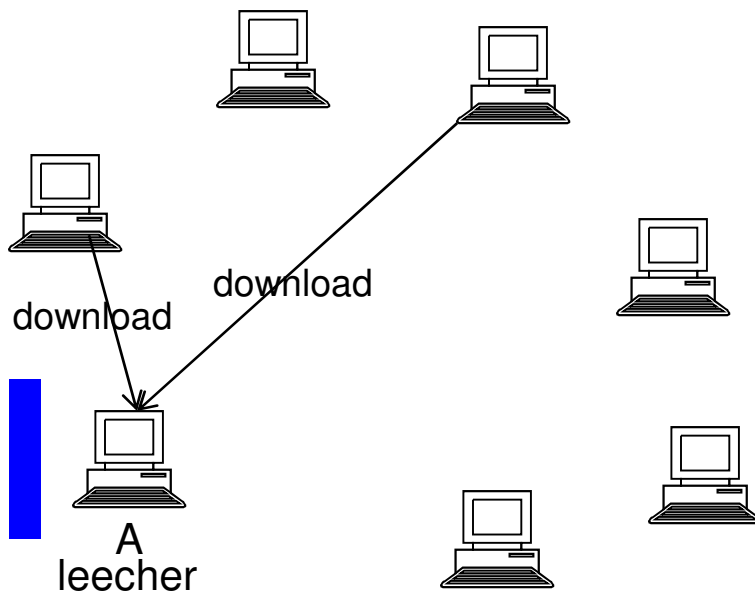
Credo's system architecture



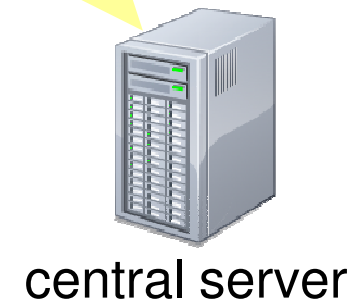
- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils



Credo's system architecture

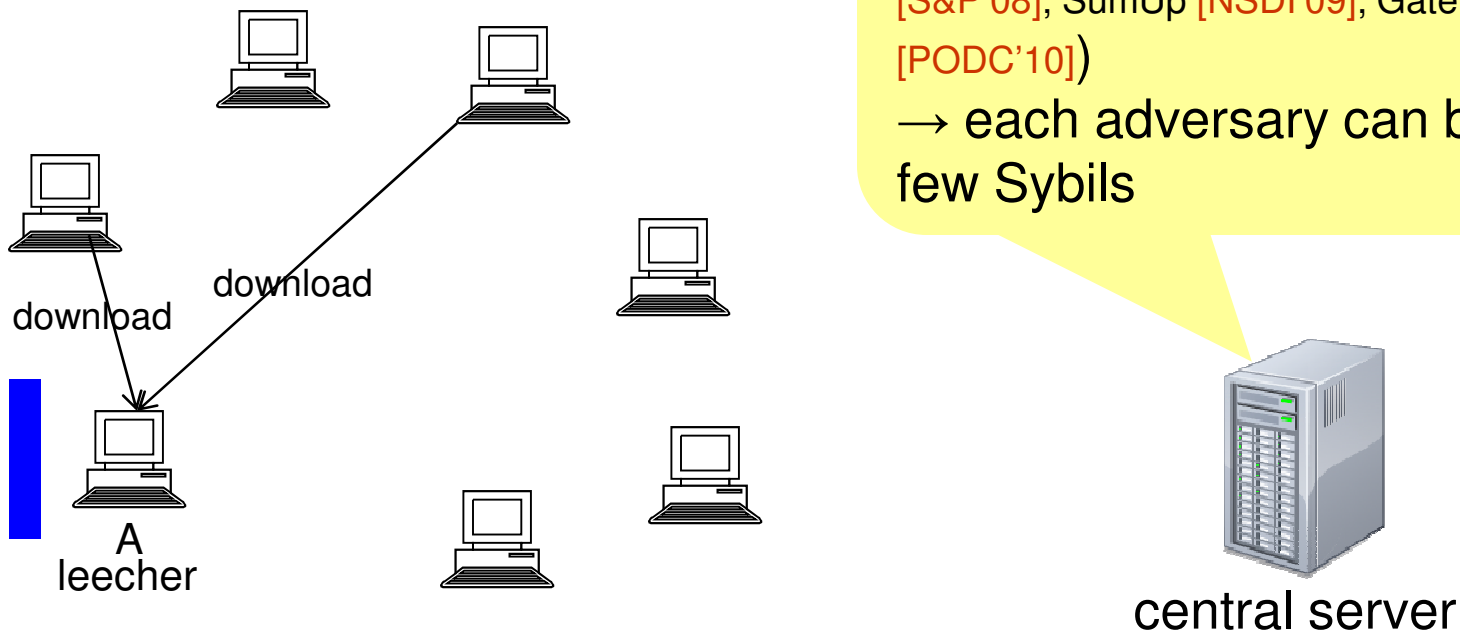


- Sybil-resilient node admission using social network (SybilLimit [S&P'08], SumUp [NSDI'09], GateKeeper [PODC'10])
→ each adversary can bring in few Sybils



$$\text{Rep} = (\# \text{ uploads}) - (\# \text{ downloads})$$

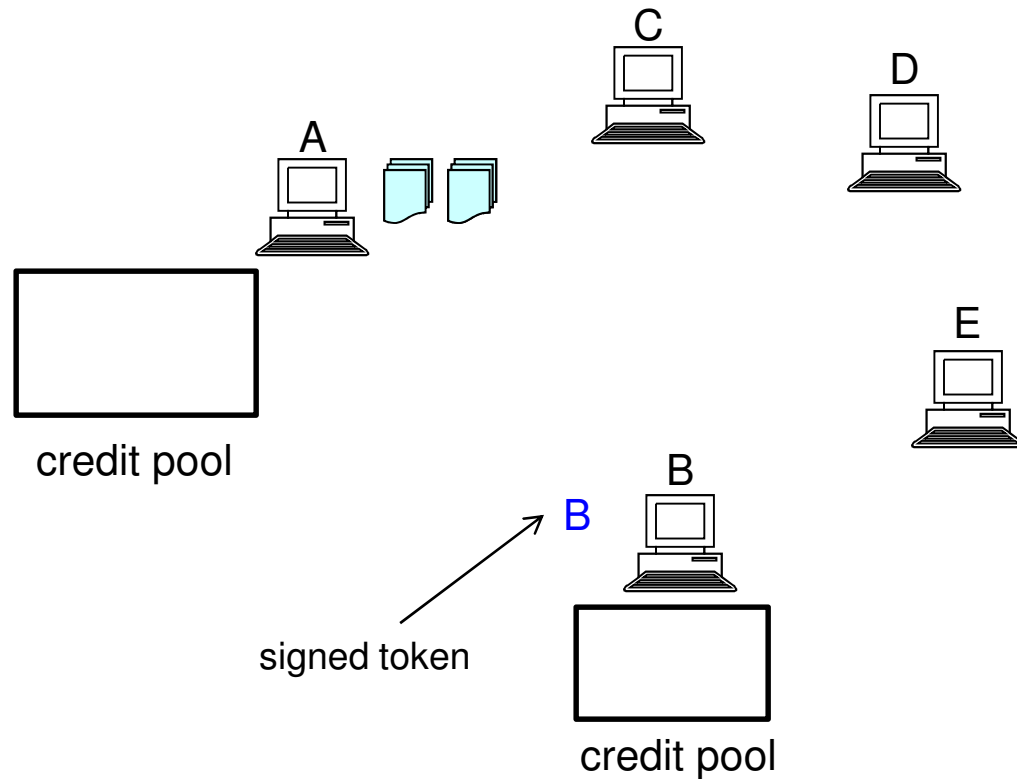
Credo's system architecture



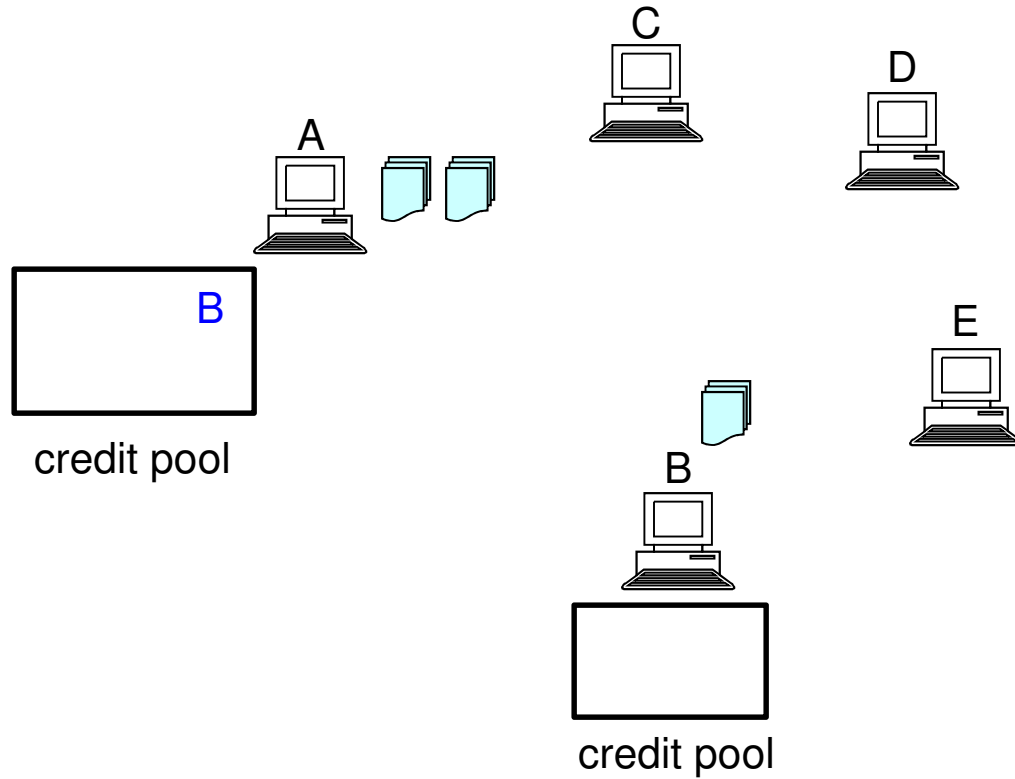
$$\text{Rep} = (\# \text{ uploads}) - (\# \text{ downloads})$$

Seeders choose the highest reputation leecher to serve

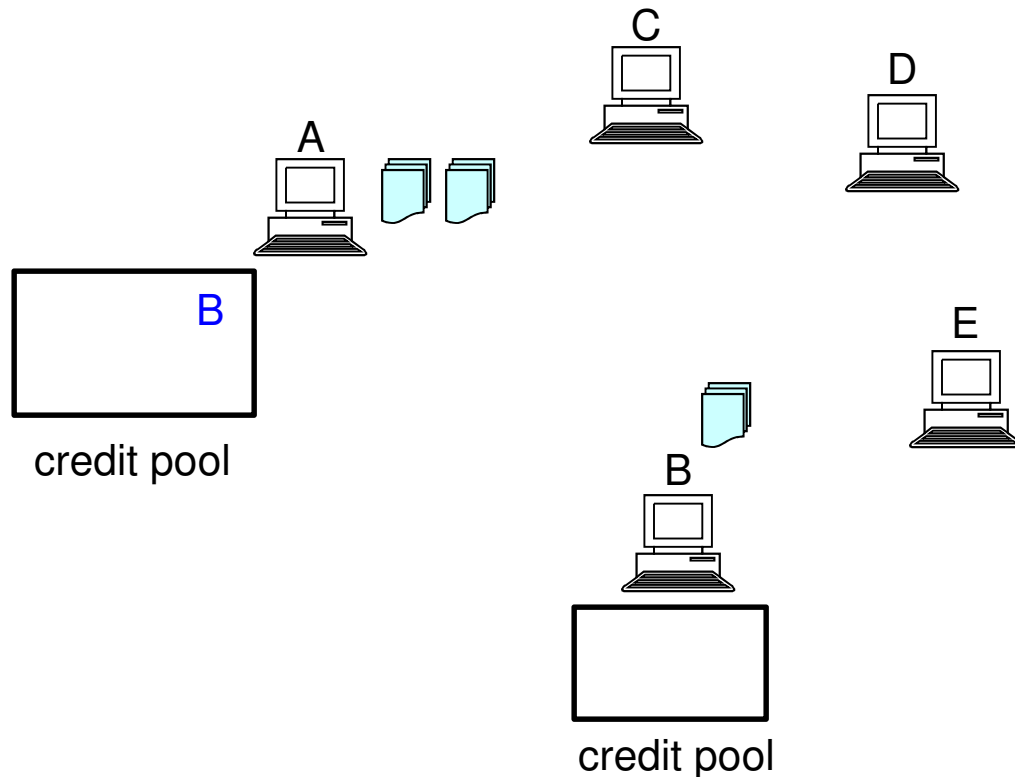
Seeders collect credits in exchange for uploads



Nodes issue their own credits

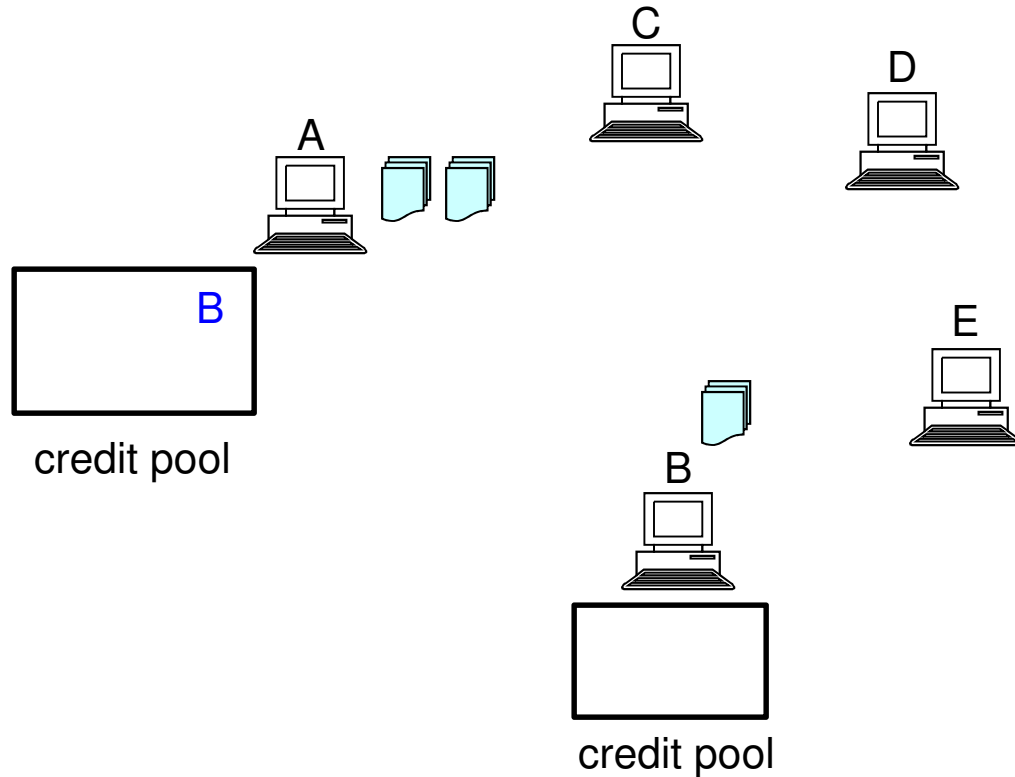


Nodes issue their own credits



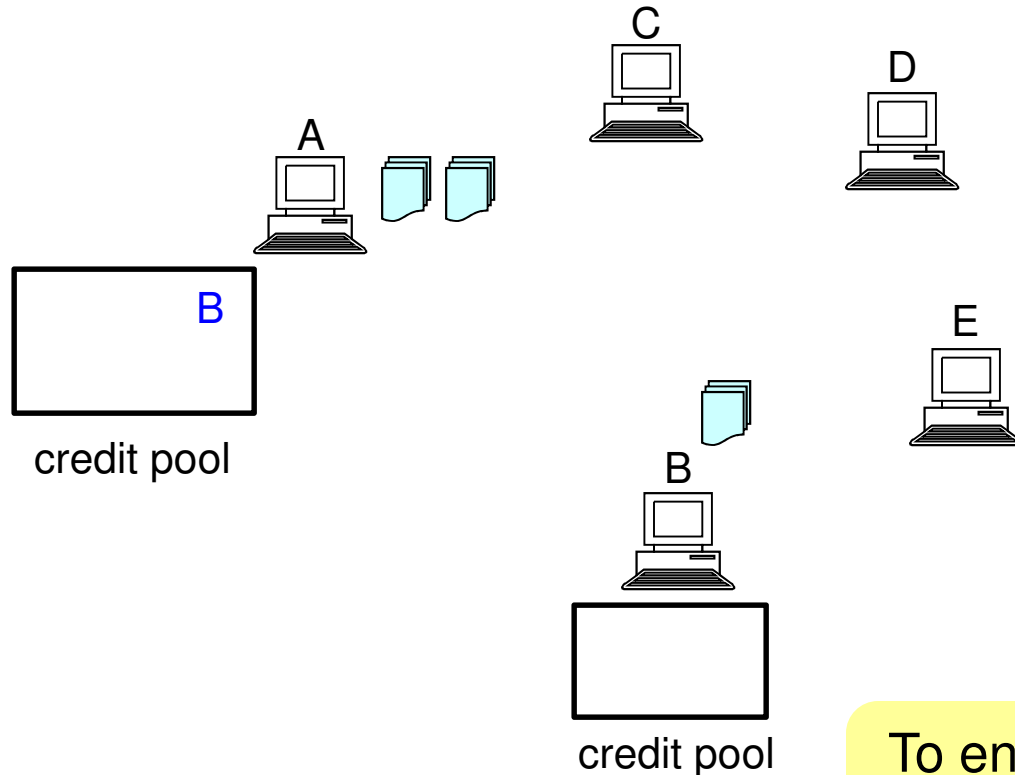
$$\text{Rep} = (\# \text{ credit earned}) - (\# \text{ issued credit}) \quad 19$$

Nodes issue their own credits



$$\text{Rep} = (\# \text{ credit earned}) - 2 \cdot (\# \text{ issued credit})_{20}$$

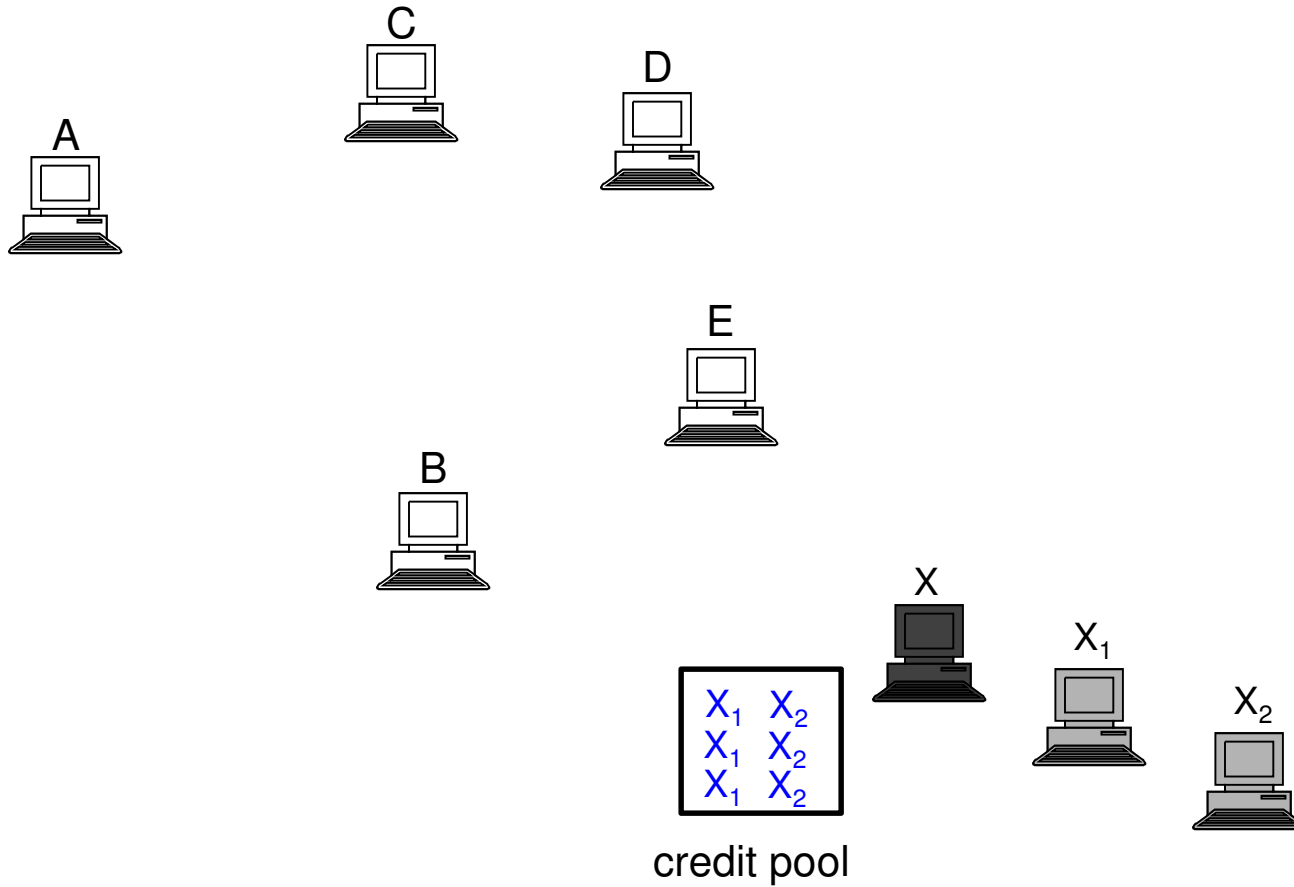
Nodes issue their own credits



To encourage nodes to use credits in credit pools before issuing new credits

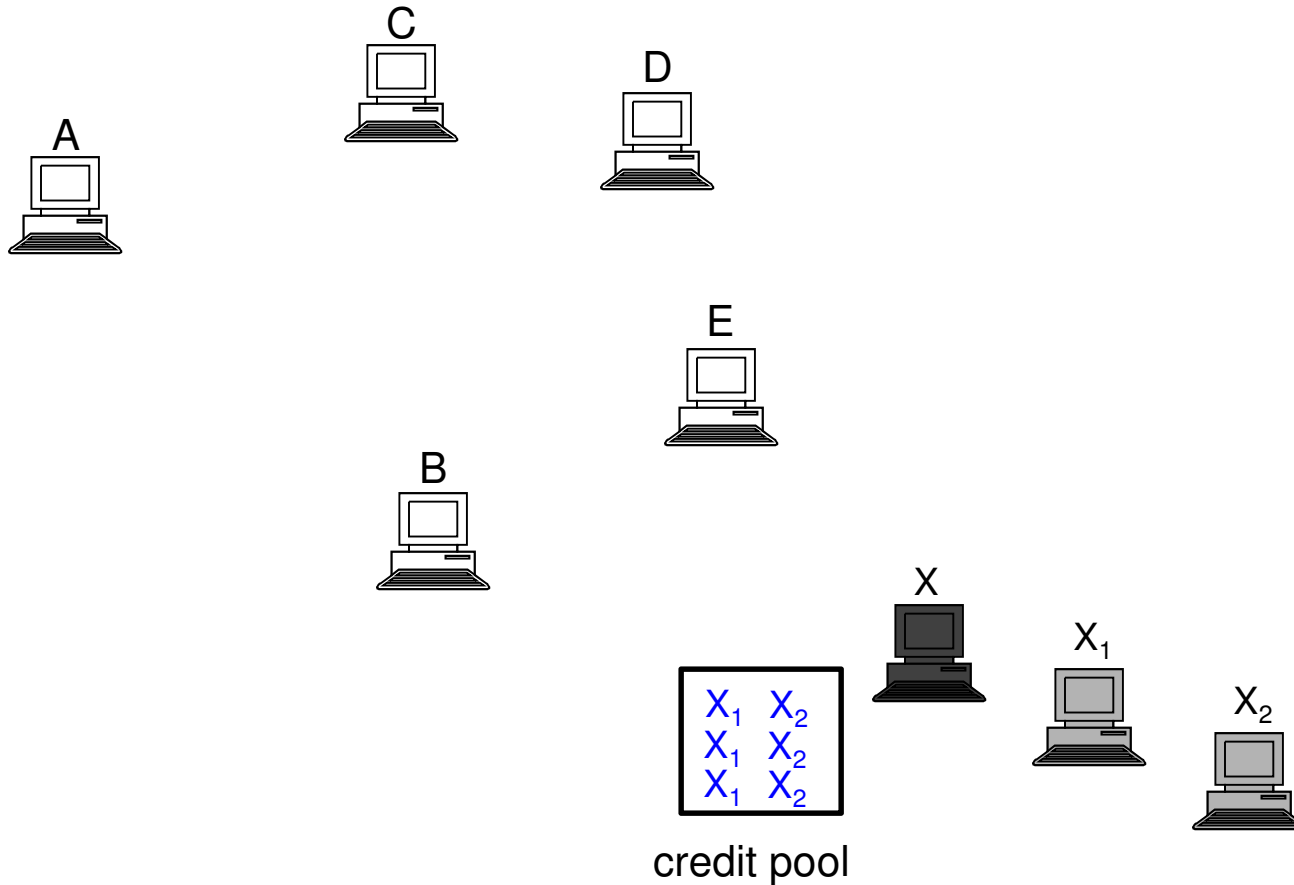
$$\text{Rep} = (\# \text{ credit earned}) - 2 \cdot (\# \text{ issued credit})_{21}$$

Sybil attack



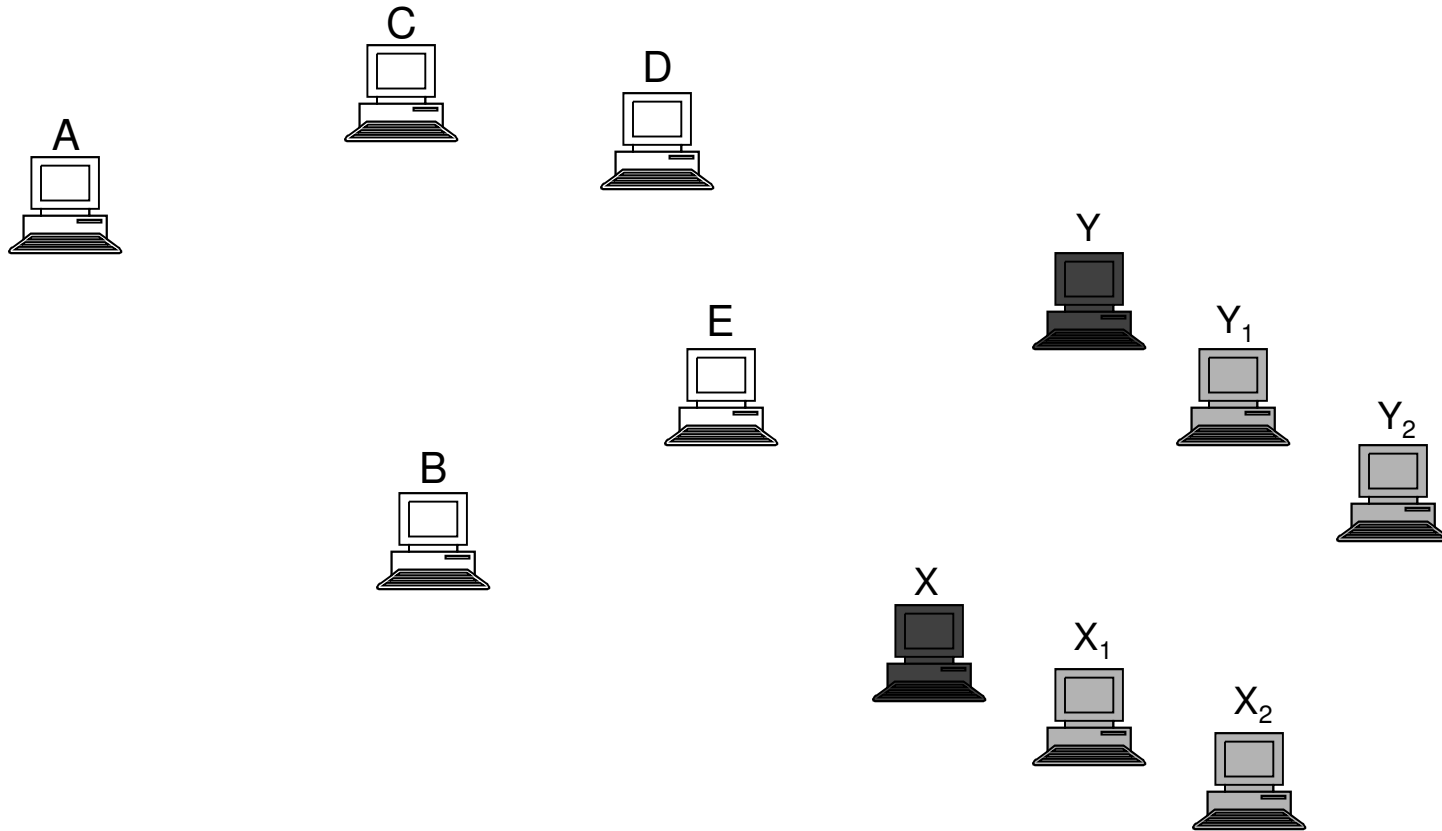
$$\text{Rep} = (\# \text{ credit earned}) - 2 \cdot (\# \text{ issued credit}) \quad 22$$

Idea 1: Credit diversity



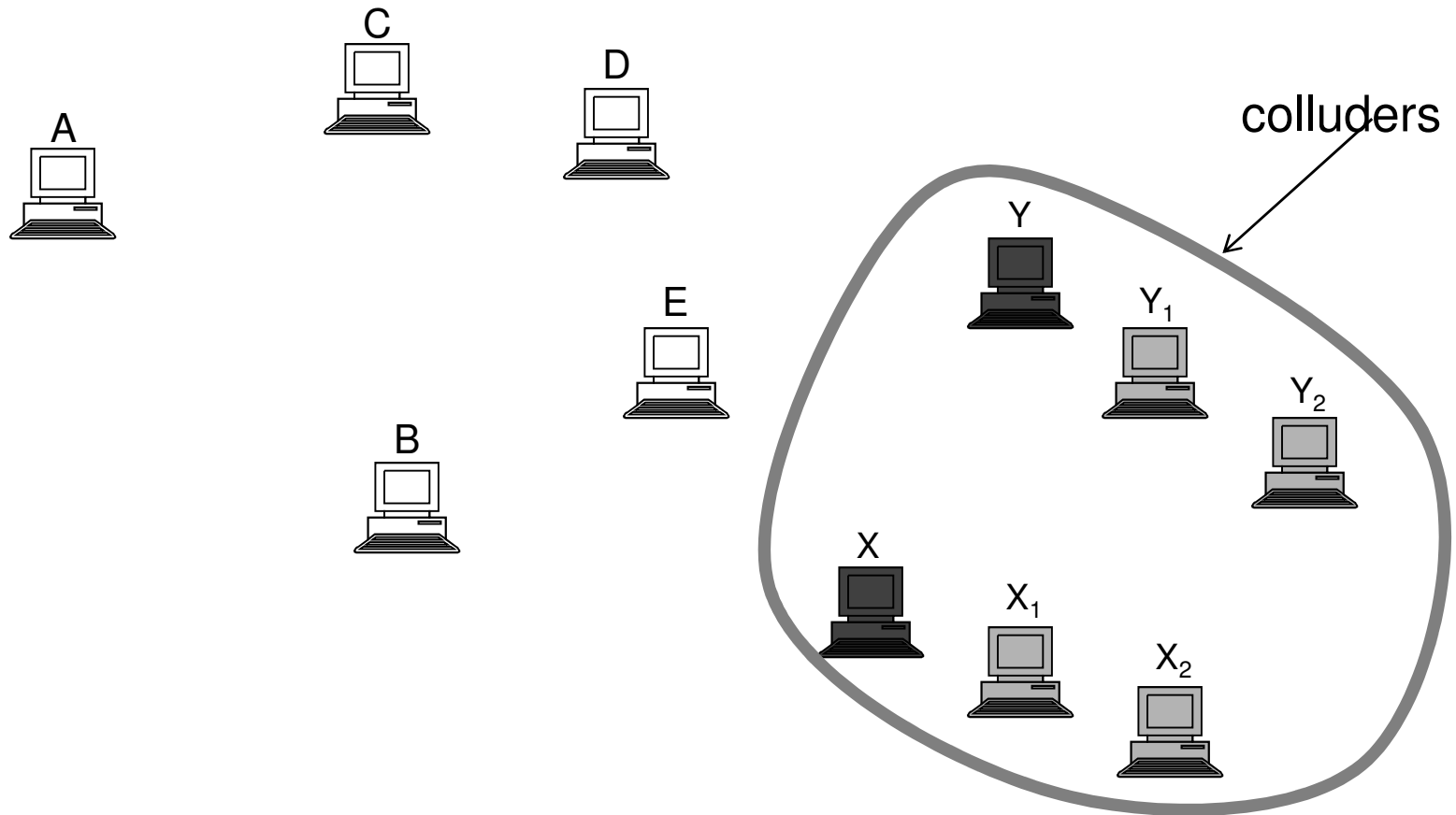
$$\text{Rep} = (\# \text{ different issuers}) - 2 \cdot (\# \text{ issued credit})_{23}$$

Credit diversity is not enough



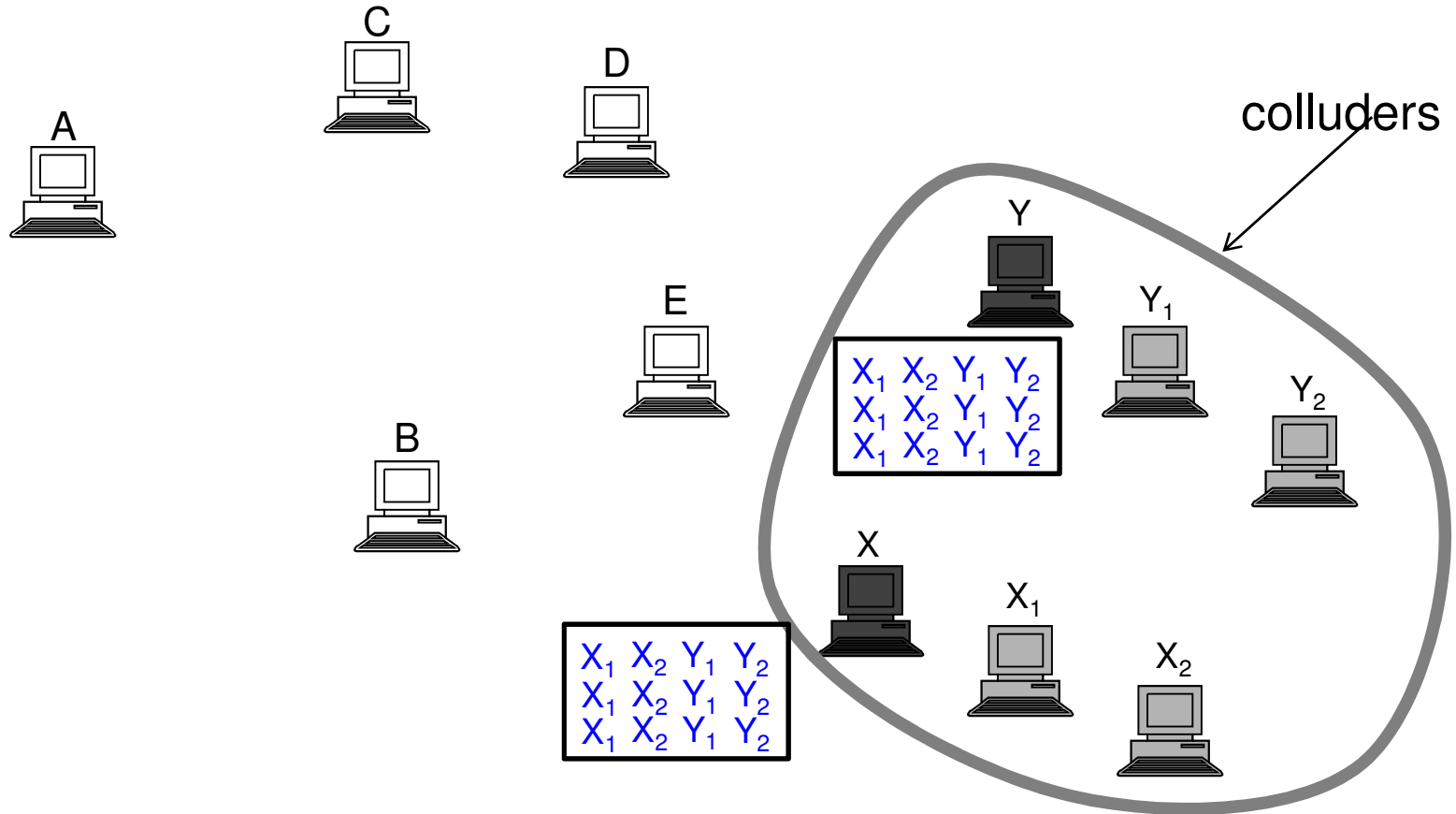
$$\text{Rep} = (\# \text{ different issuers}) - 2 \cdot (\# \text{ issued credit})_{24}$$

Credit diversity is not enough



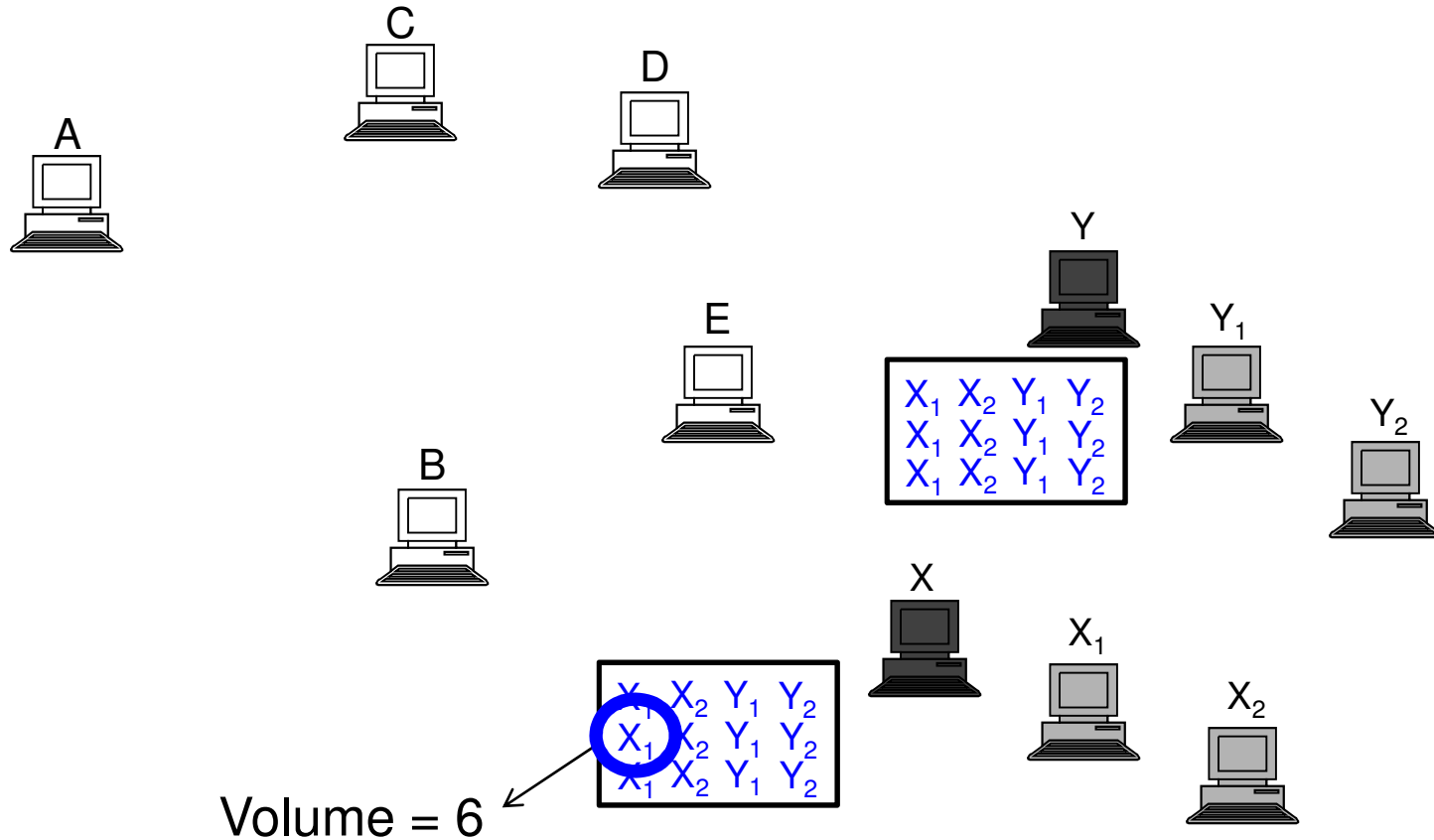
$$\text{Rep} = (\# \text{ different issuers}) - 2 \cdot (\# \text{ issued credit})_{25}$$

Credit diversity is not enough



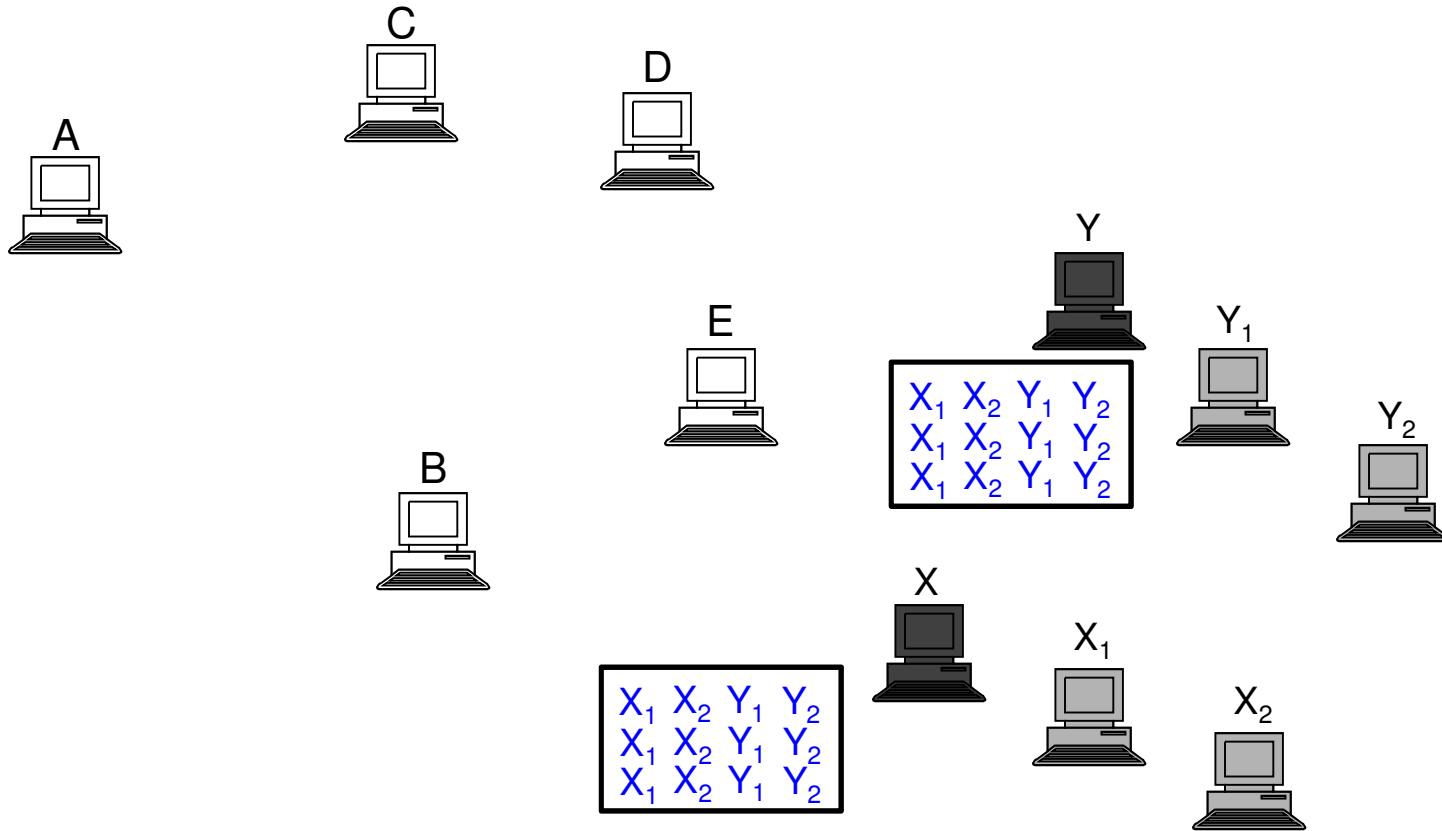
$$\text{Rep} = (\# \text{ different issuers}) - 2 \cdot (\# \text{ issued credit})_{26}$$

Credit pool of attackers vs honest nodes



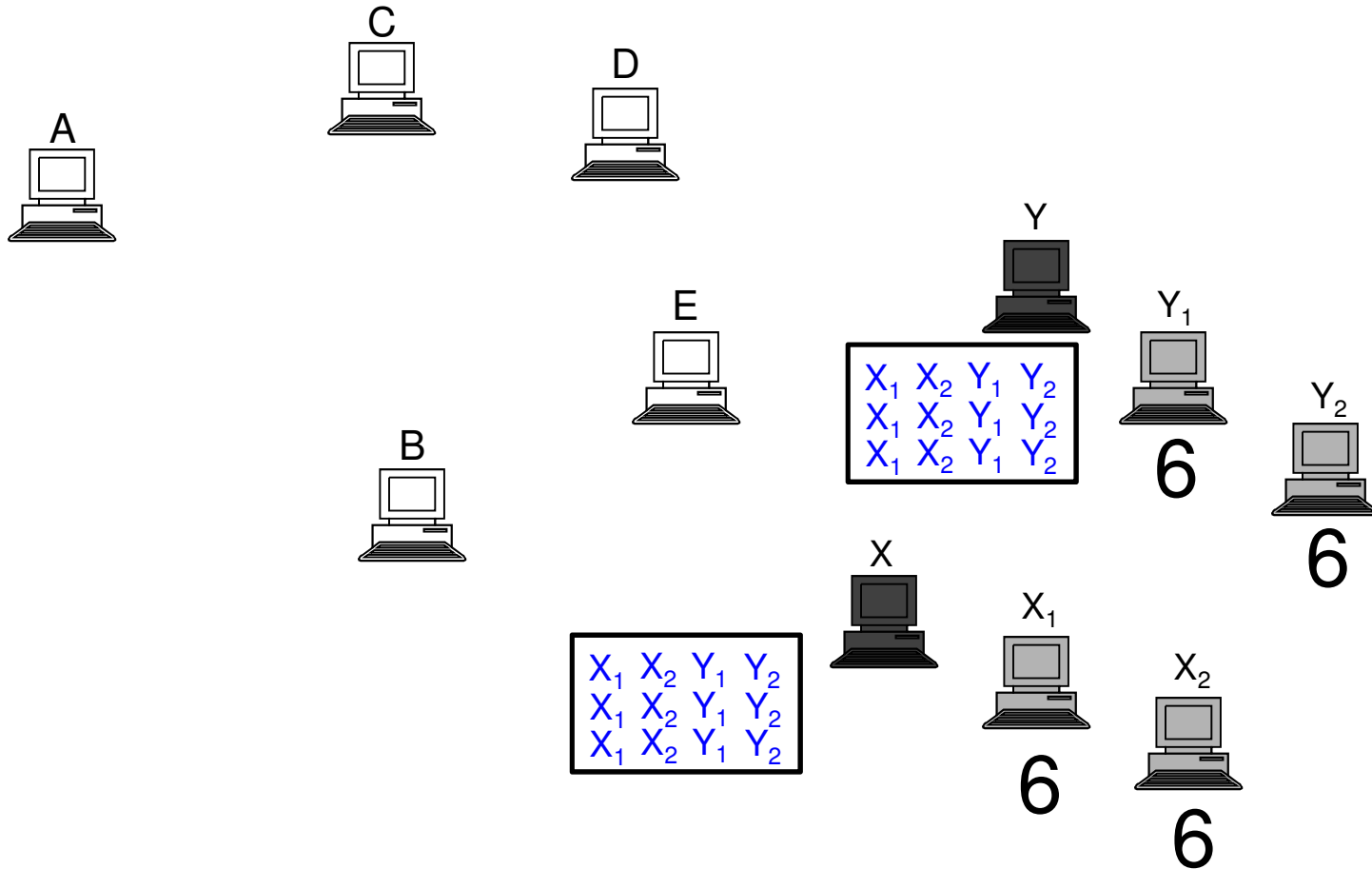
Volume(c) : # of credits issued by the issuer of c

Credit pool of attackers vs honest nodes



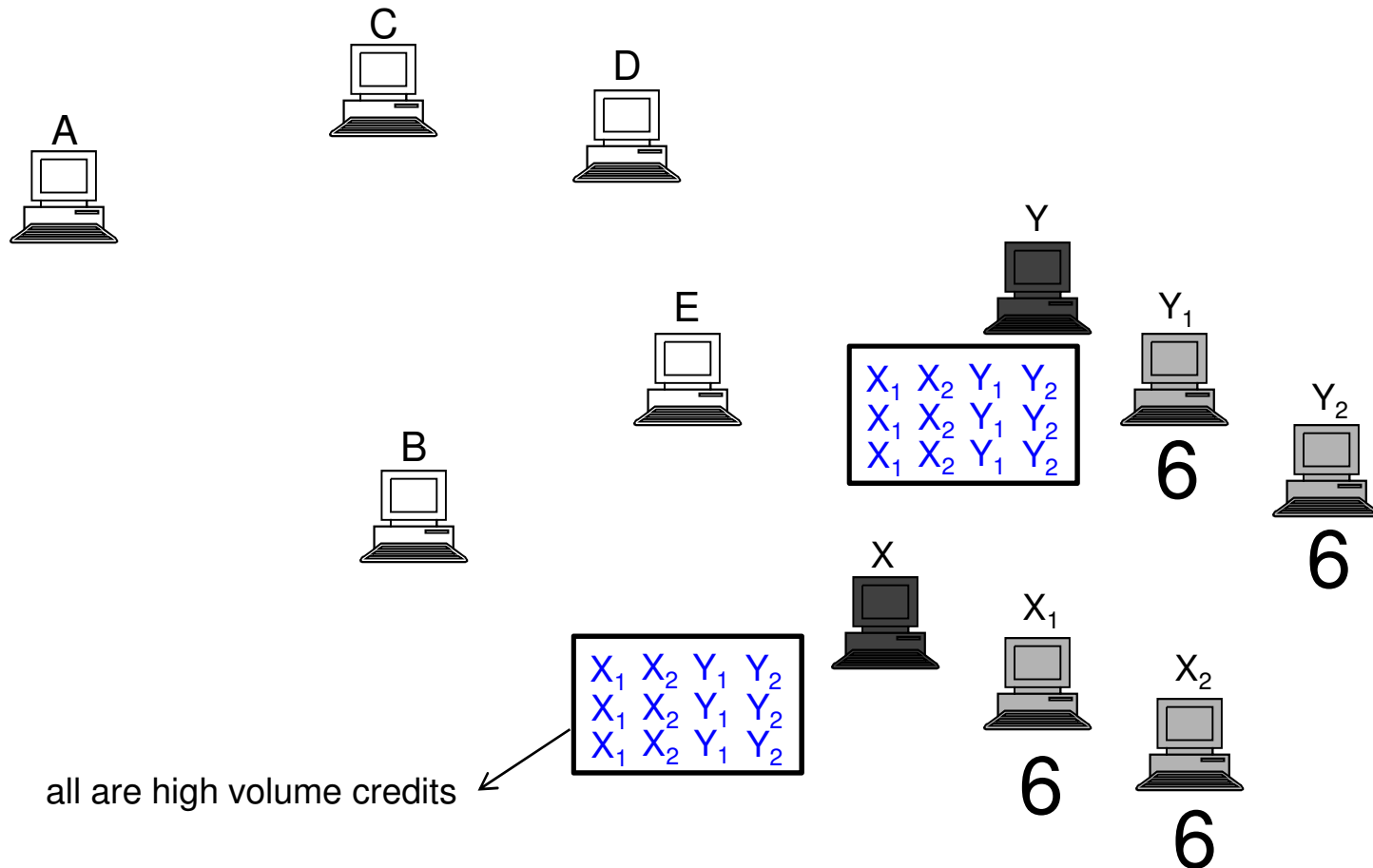
Volume(c) : # of credits issued by the issuer of c

Credit pool of attackers vs honest nodes



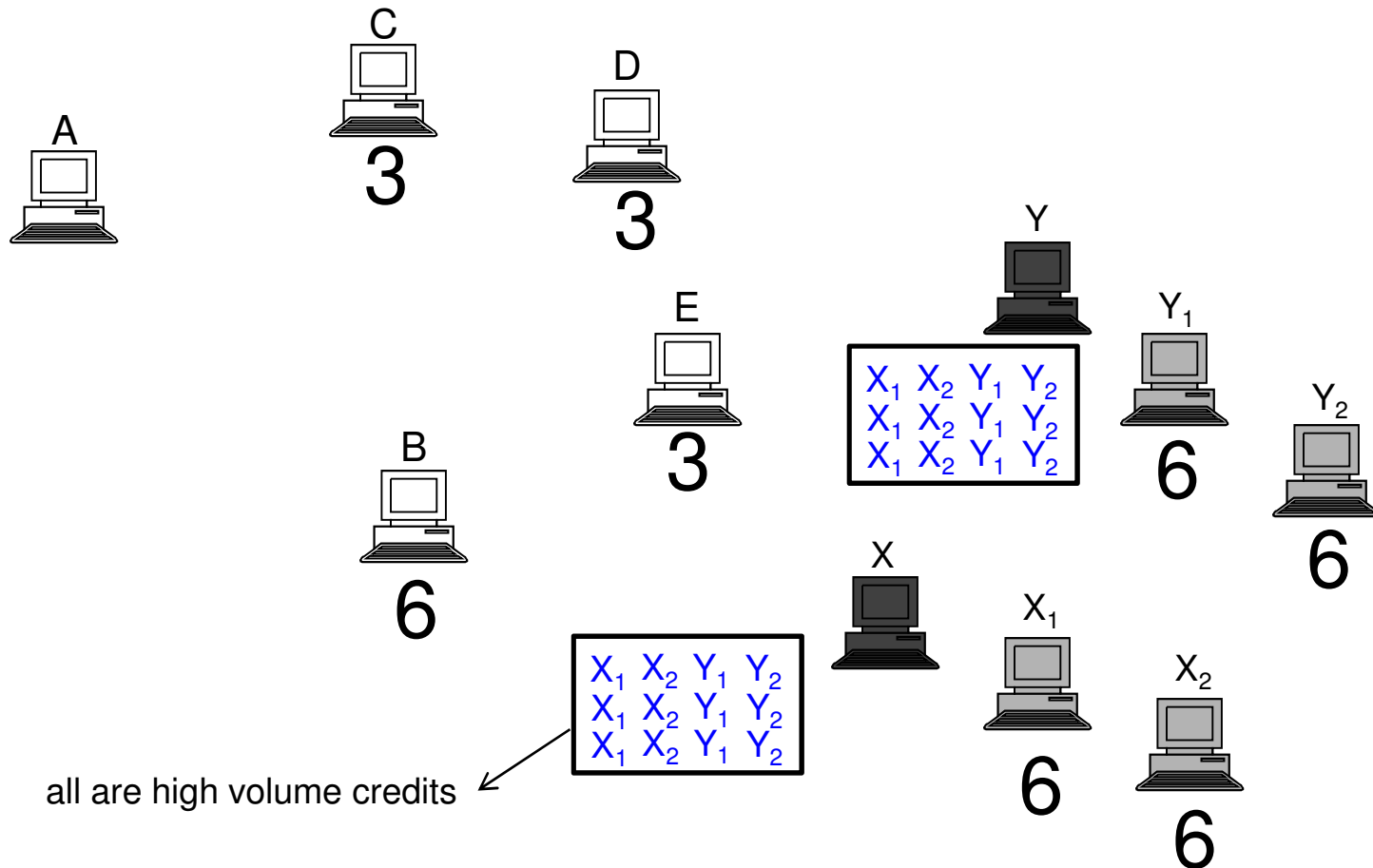
Volume(c) : # of credits issued by the issuer of c

Credit pool of attackers vs honest nodes



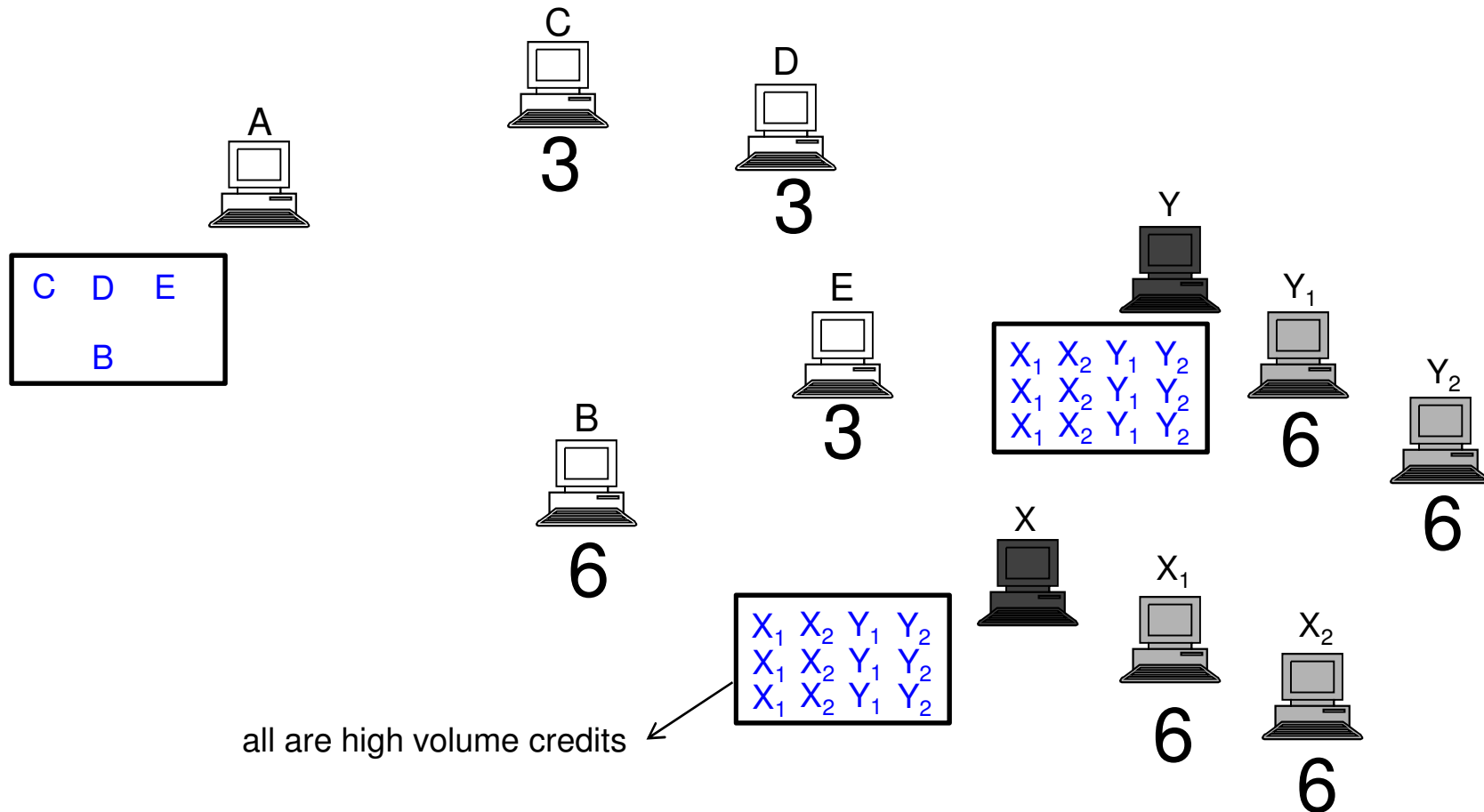
Volume(c) : # of credits issued by the issuer of c

Credit pool of attackers vs honest nodes



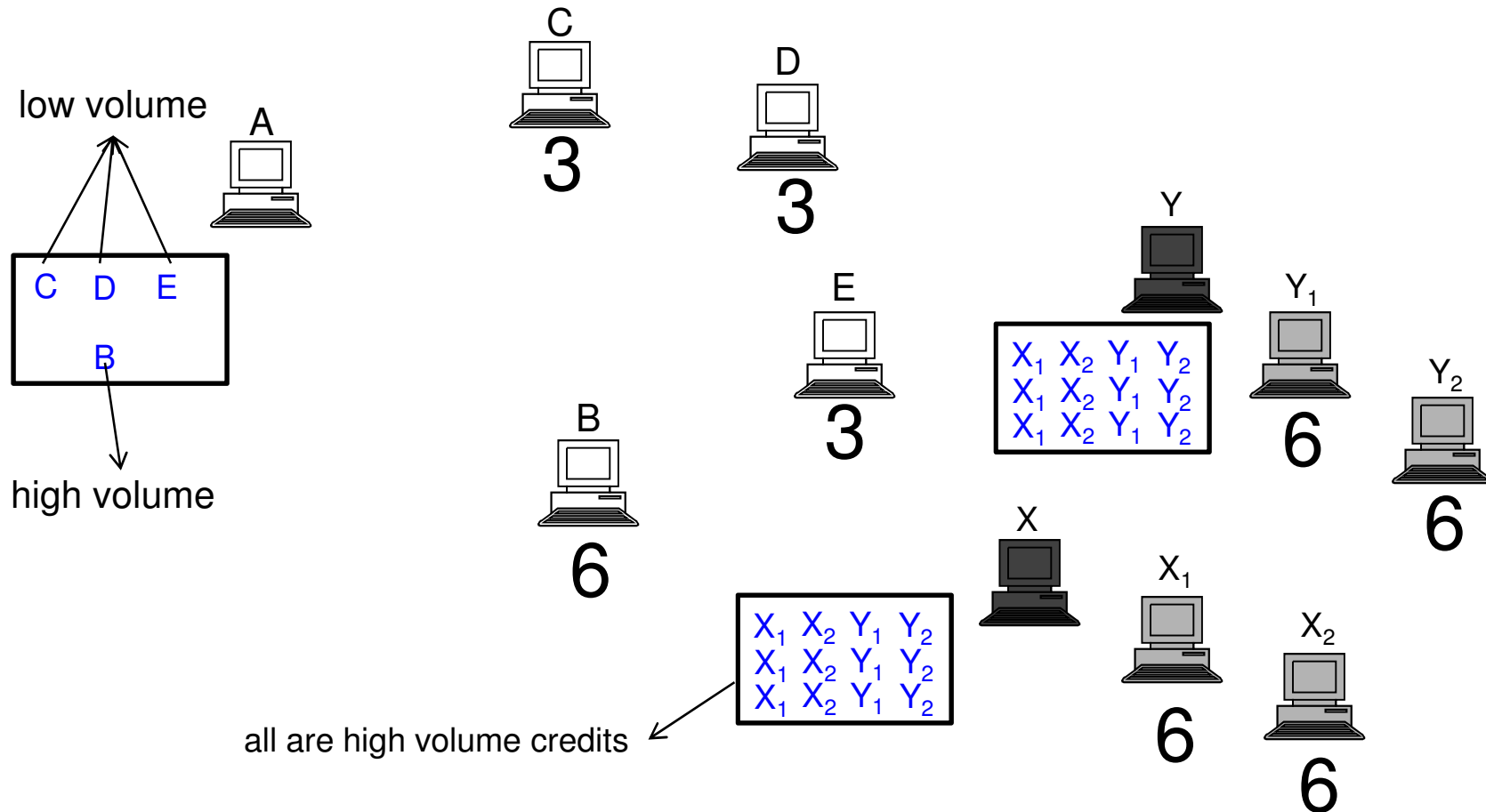
Volume(c) : # of credits issued by the issuer of c

Credit pool of attackers vs honest nodes



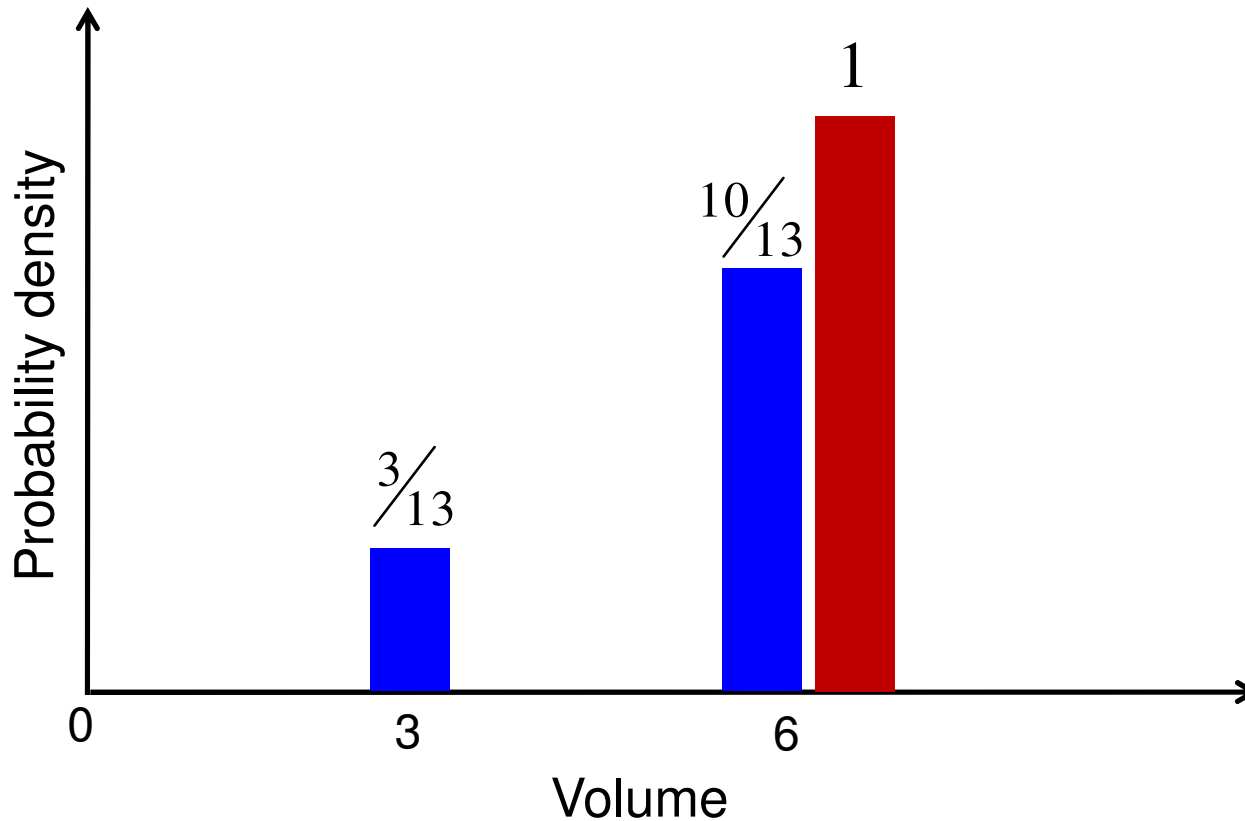
Volume(c) : # of credits issued by the issuer of c

Credit pool of attackers vs honest nodes



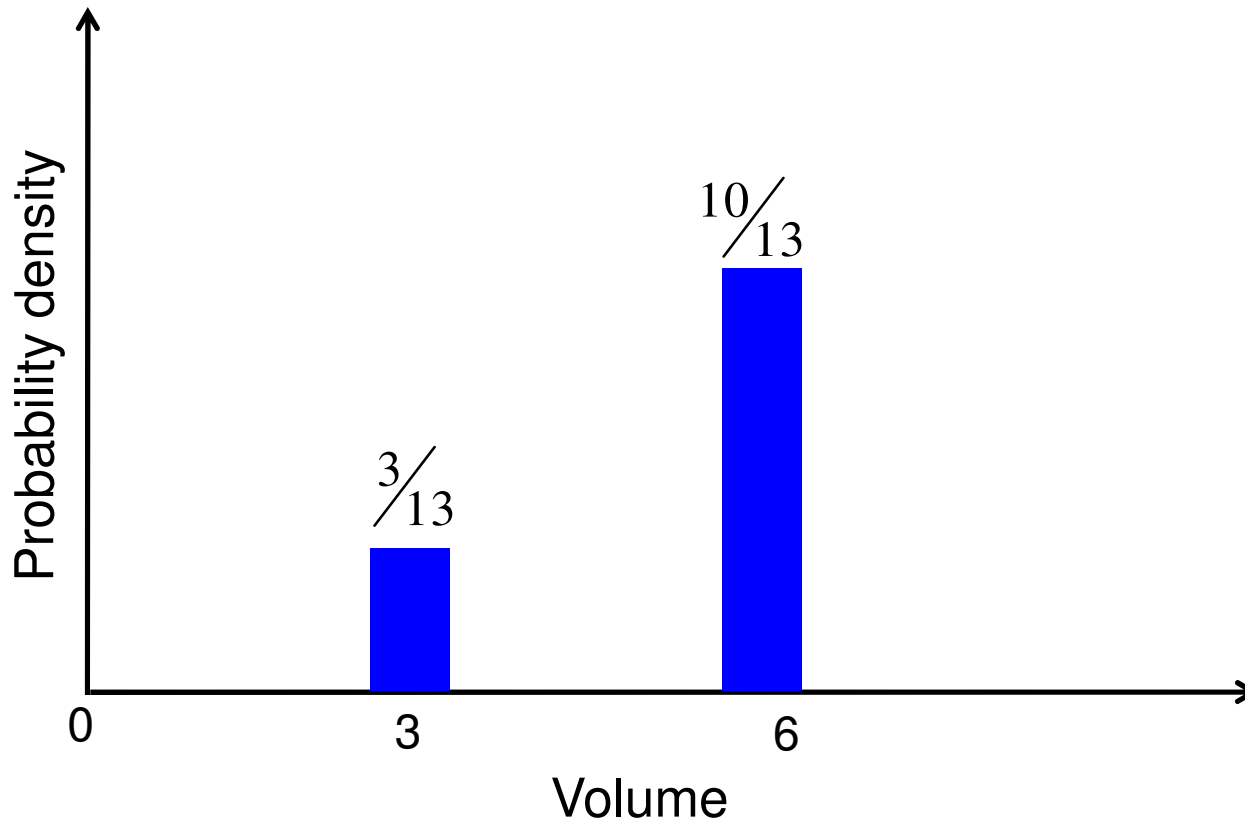
Volume(c) : # of credits issued by the issuer of c

Distribution of credits' volume



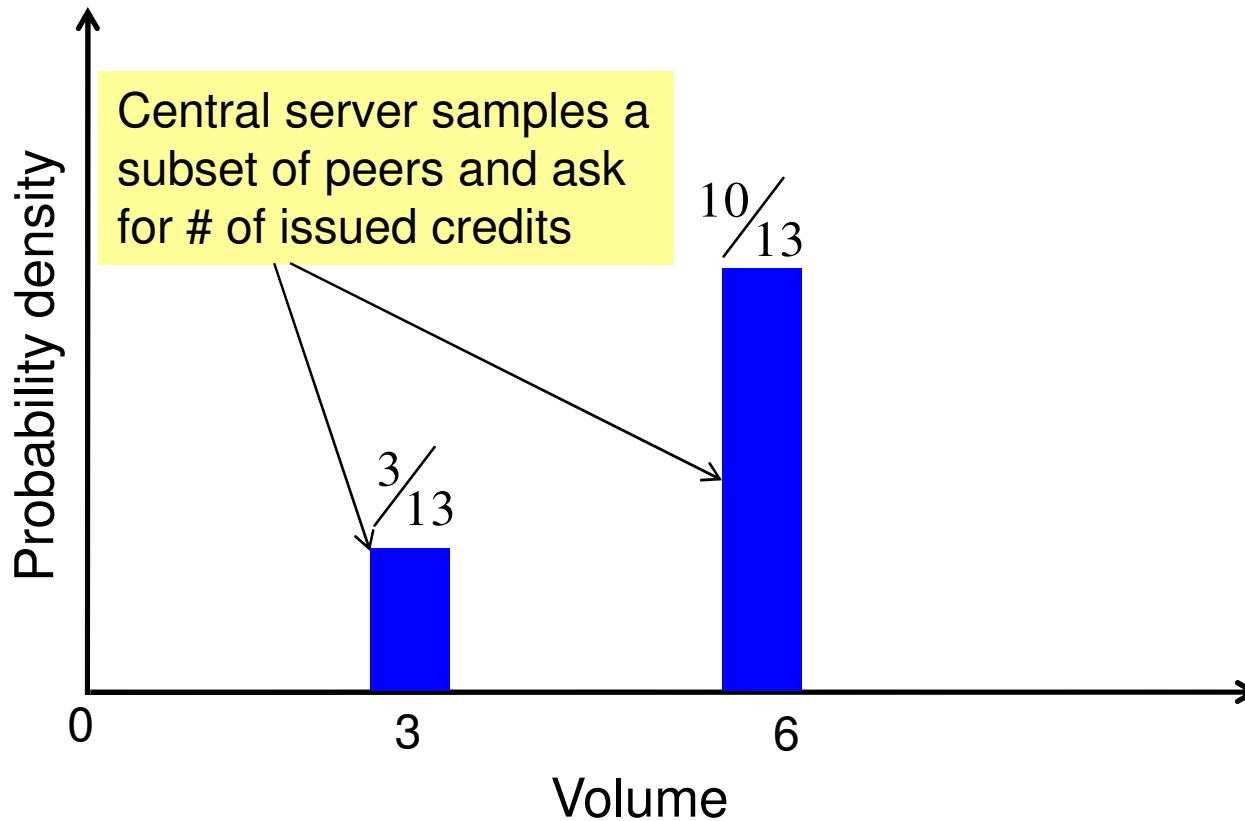
- Expected volume distribution in a normal credit pool
- Volume distribution in an adversary's credit pool

Idea 2: Modeling good behavior



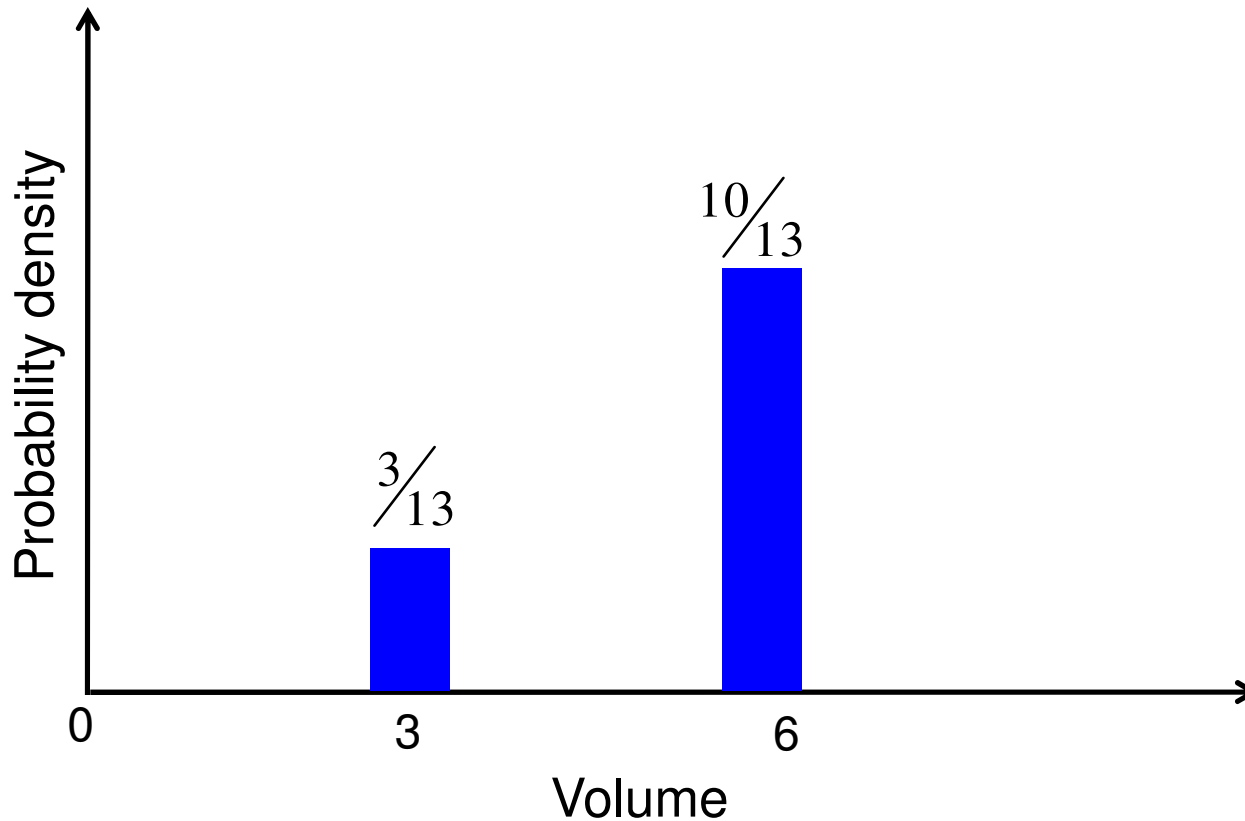
 Expected volume distribution in a normal credit pool

Idea 2: Modeling good behavior



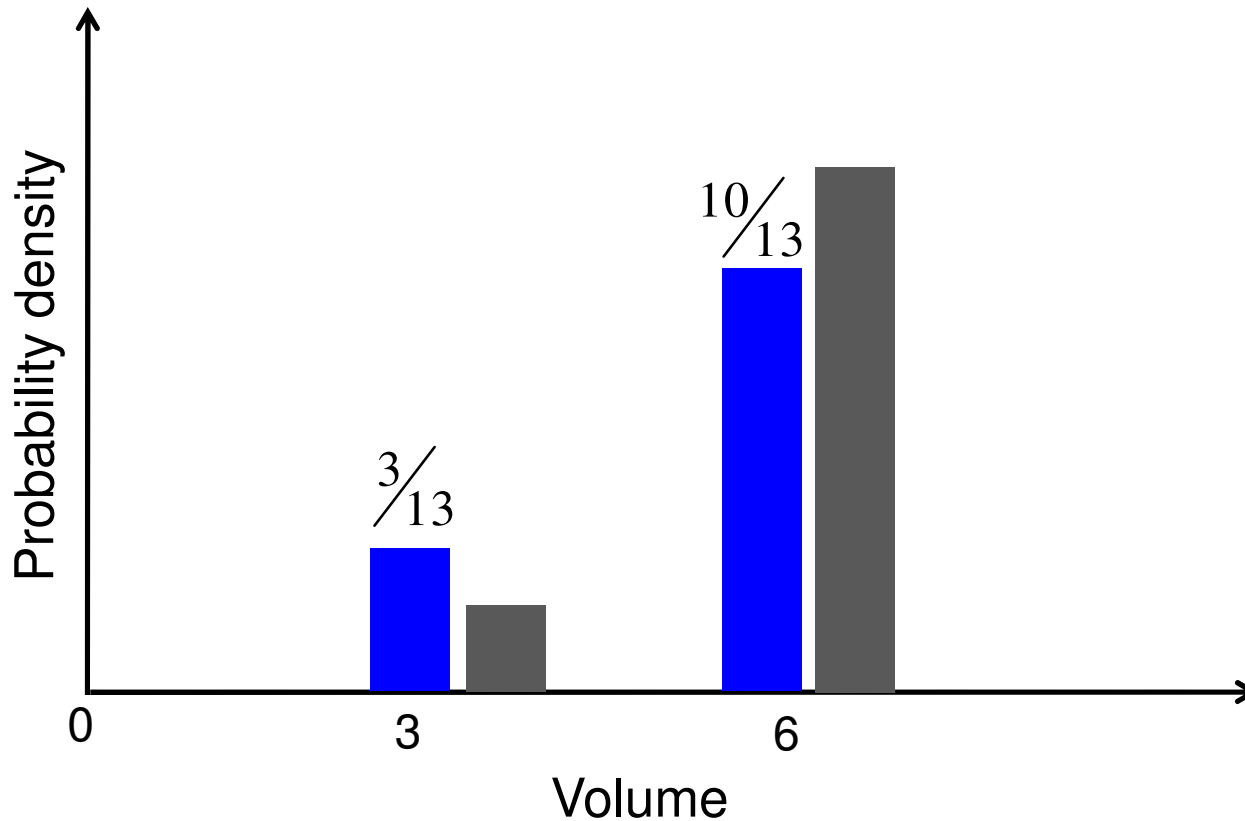
 Expected volume distribution in a normal credit pool



Idea 2: Modeling good behavior



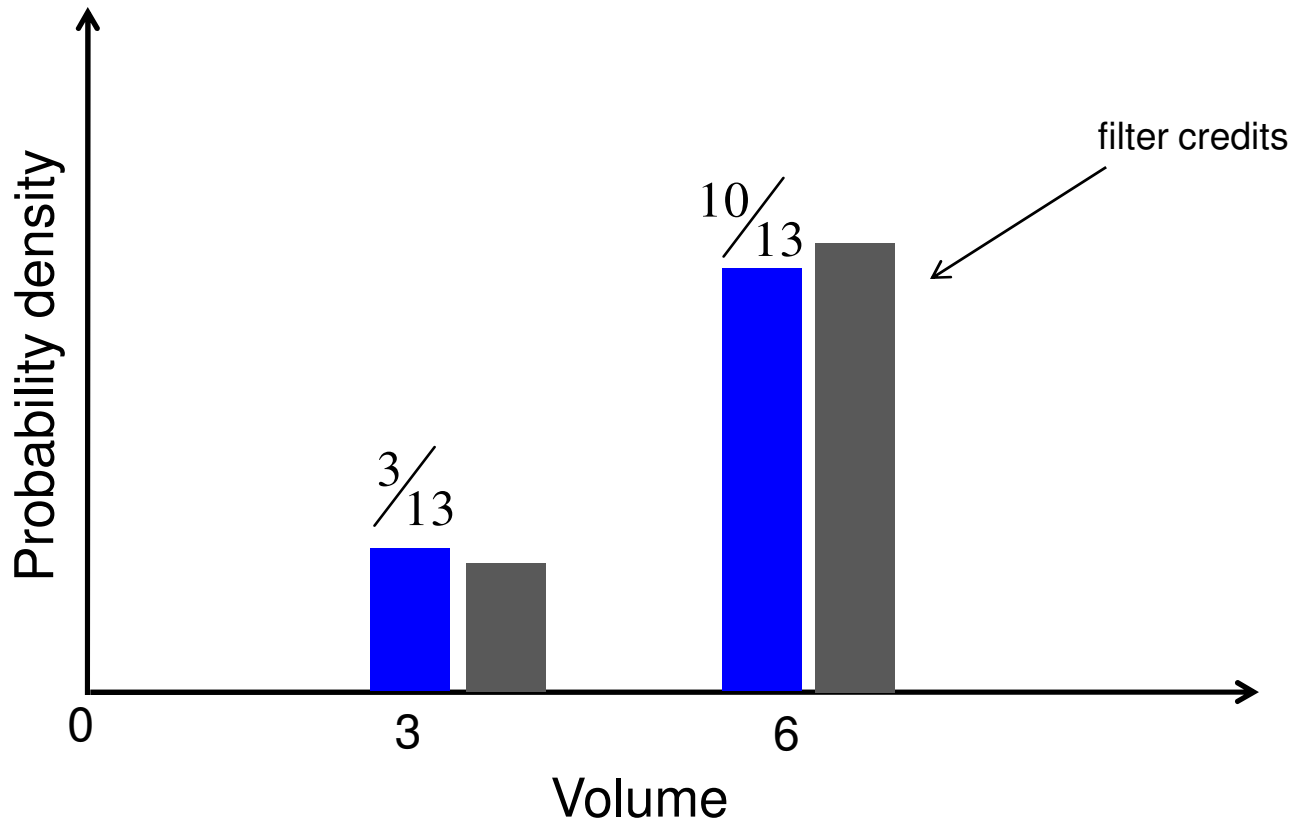
 Expected volume distribution in a normal credit pool

Idea 2: Modeling good behavior



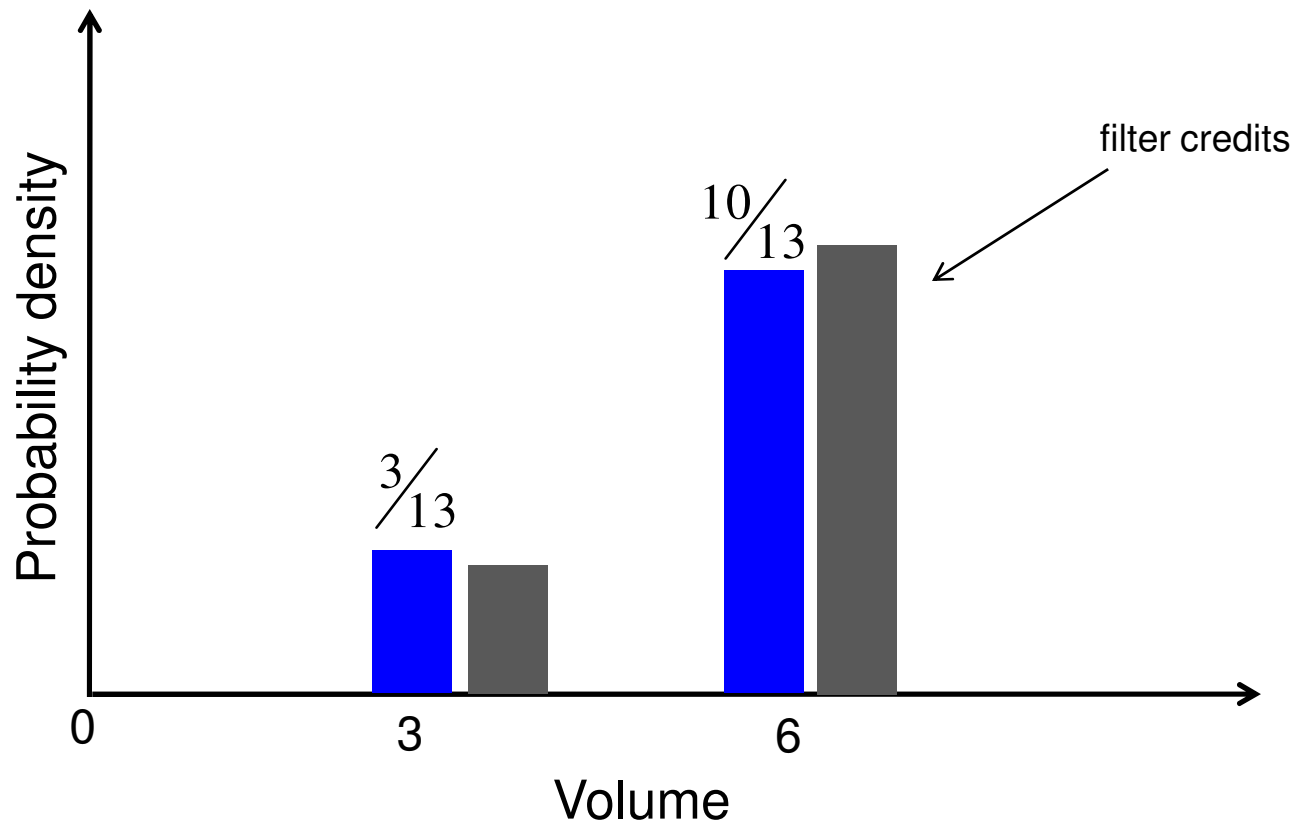
-  Expected volume distribution in a normal credit pool
-  Volume distribution in a credit pool

Idea 2: Modeling good behavior



- Expected volume distribution in a normal credit pool
- Volume distribution in a credit pool

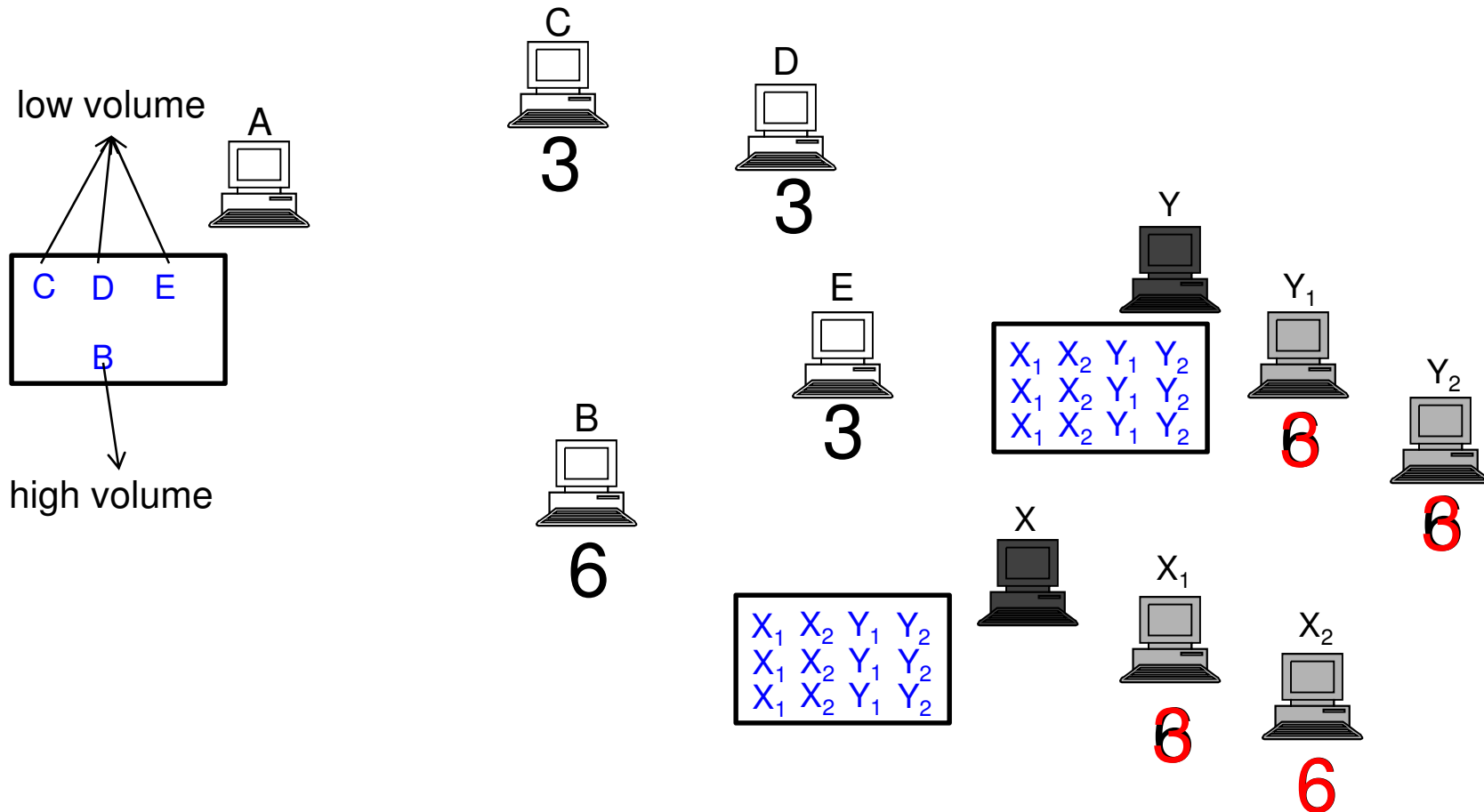
Idea 2: Modeling good behavior



- Expected volume distribution in a normal credit pool
- Volume distribution in a credit pool

$$\text{Rep} = (\text{diversity of filtered pool}) - 2 \cdot (\# \text{ issued credit}) \quad 40$$

Effect on attackers



Sybil nodes issue similar amount of credits as honest nodes

Credo's security properties

- Suppose there are k adversaries, each brings in s Sybils. They form a collusion size of $C = k \cdot s$, and do not contribute.

Credo's security properties

- Suppose there are k adversaries, each brings in s Sybils. They form a collusion size of $C = k \cdot s$, and do not contribute.
 - The reputation of each adversary is bounded by the collusion size C

Credo's security properties

- Suppose there are k adversaries, each brings in s Sybils. They form a collusion size of $C = k \cdot s$, and do not contribute.
 - The reputation of each adversary is bounded by the collusion size C
 - Reputation of adversary decrease after $\gamma \cdot (s \cdot x)$ download

Credo's security properties

- Suppose there are k adversaries, each brings in s Sybils. They form a collusion size of $C = k \cdot s$, and do not contribute.
 - The reputation of each adversary is bounded by the collusion size C
 - Reputation of adversary decrease after $\gamma \cdot (s \cdot x)$ download

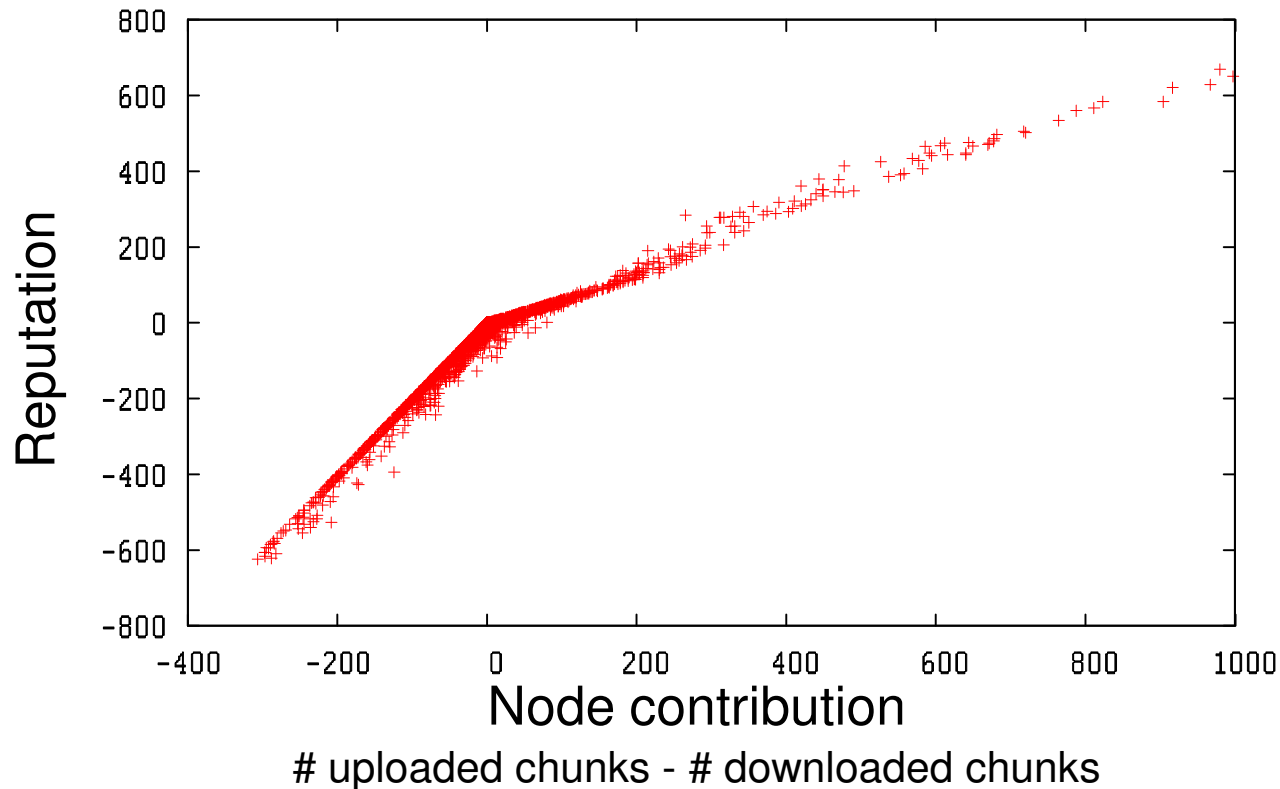
A small constant

Average number of self-issued credits of an issuer

Auditing to catch misbehavior

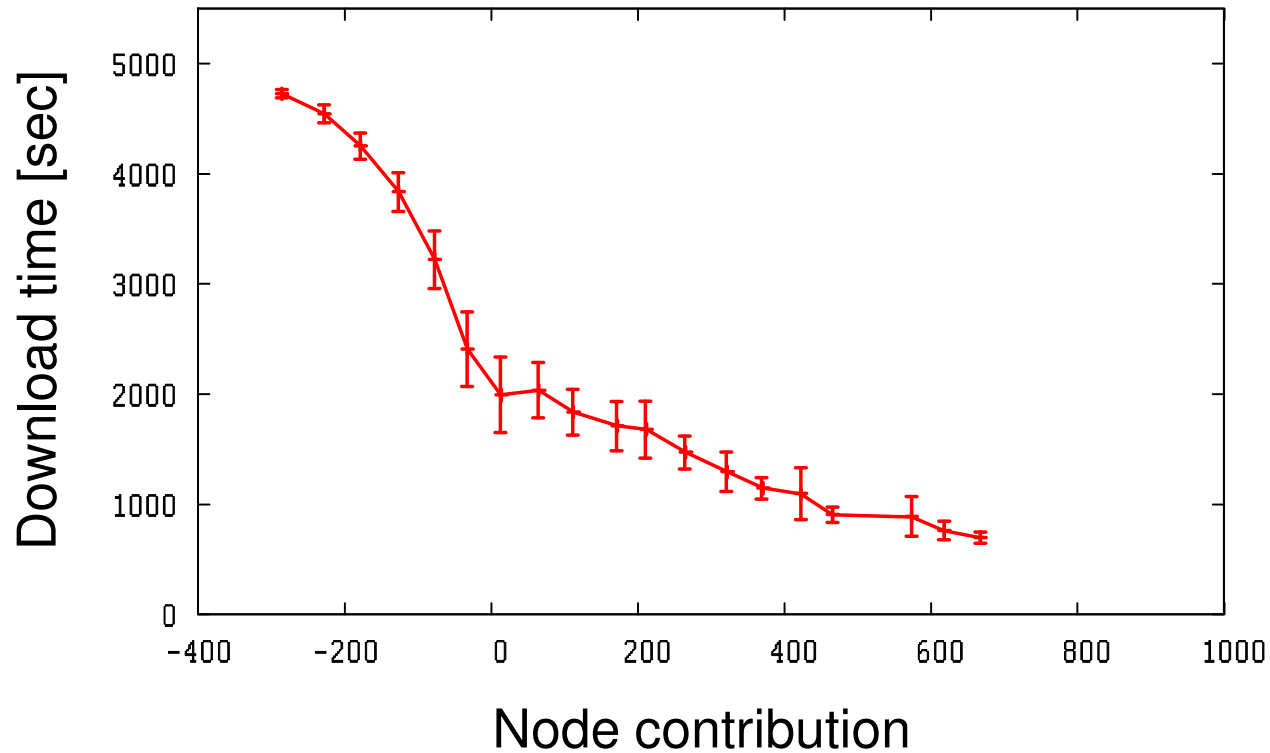
- Nodes can lie
 - Double spend credits
 - Falsely report number of issued credits
 - Many others ...
- Audit to catch liars with provable evidence (PeerReview) → disincentivize nodes to lie

Credo reputation reflects node contribution

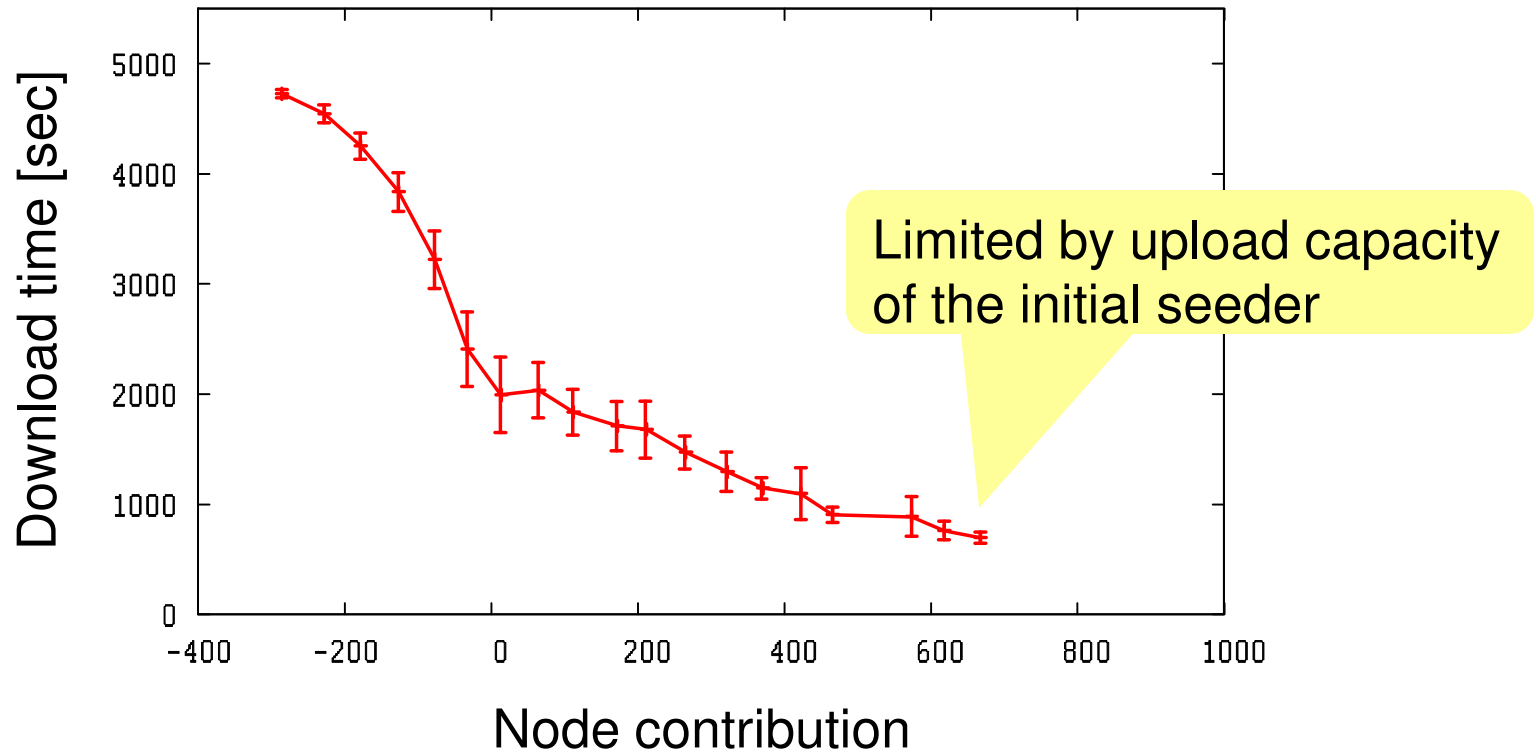


- Simulate 1 year of 3000 nodes network
- Continuously inject 100MB file and choose 300 nodes to download
- Use Maze data (2005) to model nodes' demand
- Use BitTorrent data (2007) to model nodes' upload capacity

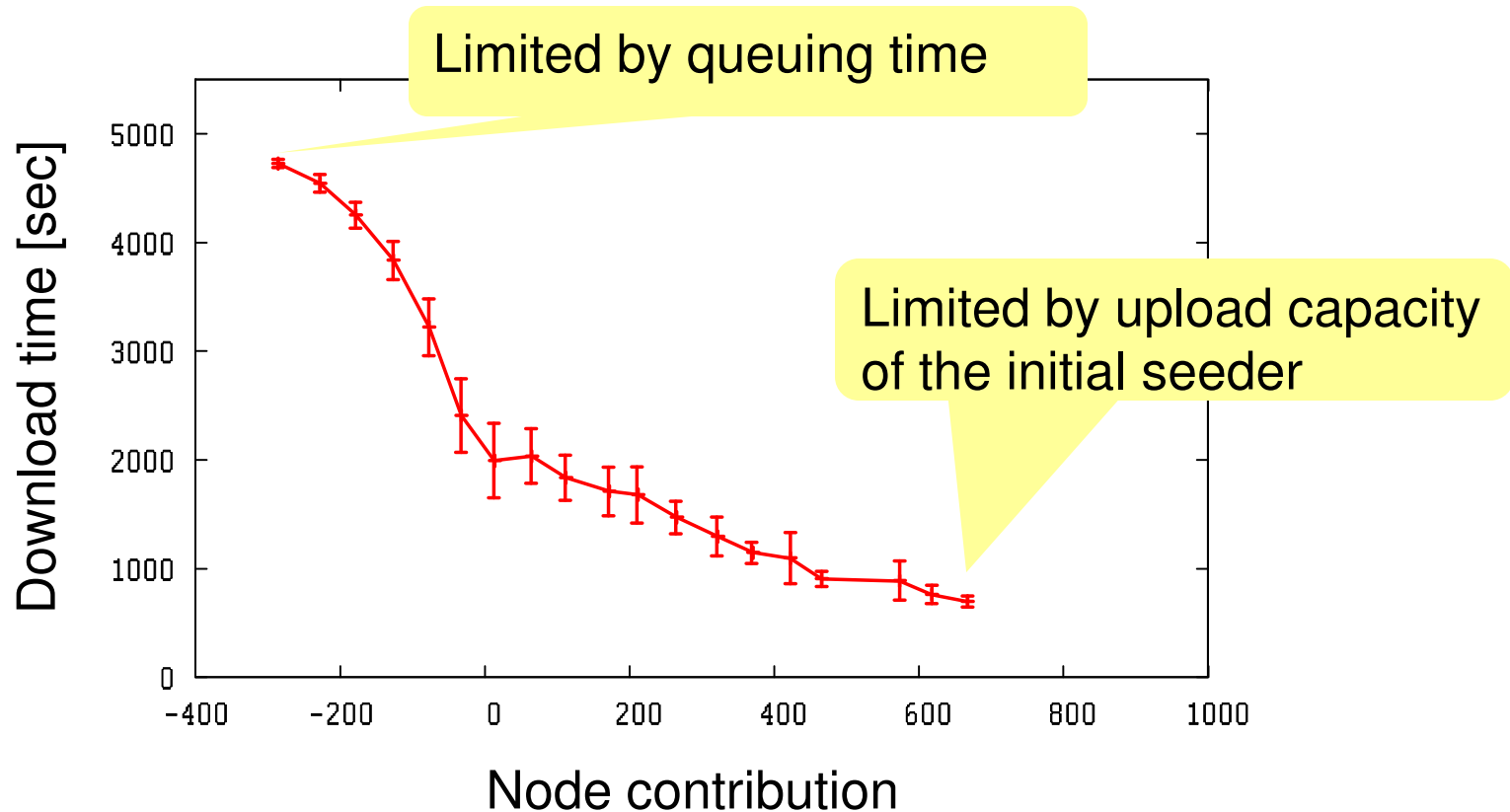
Higher reputation \rightarrow faster download



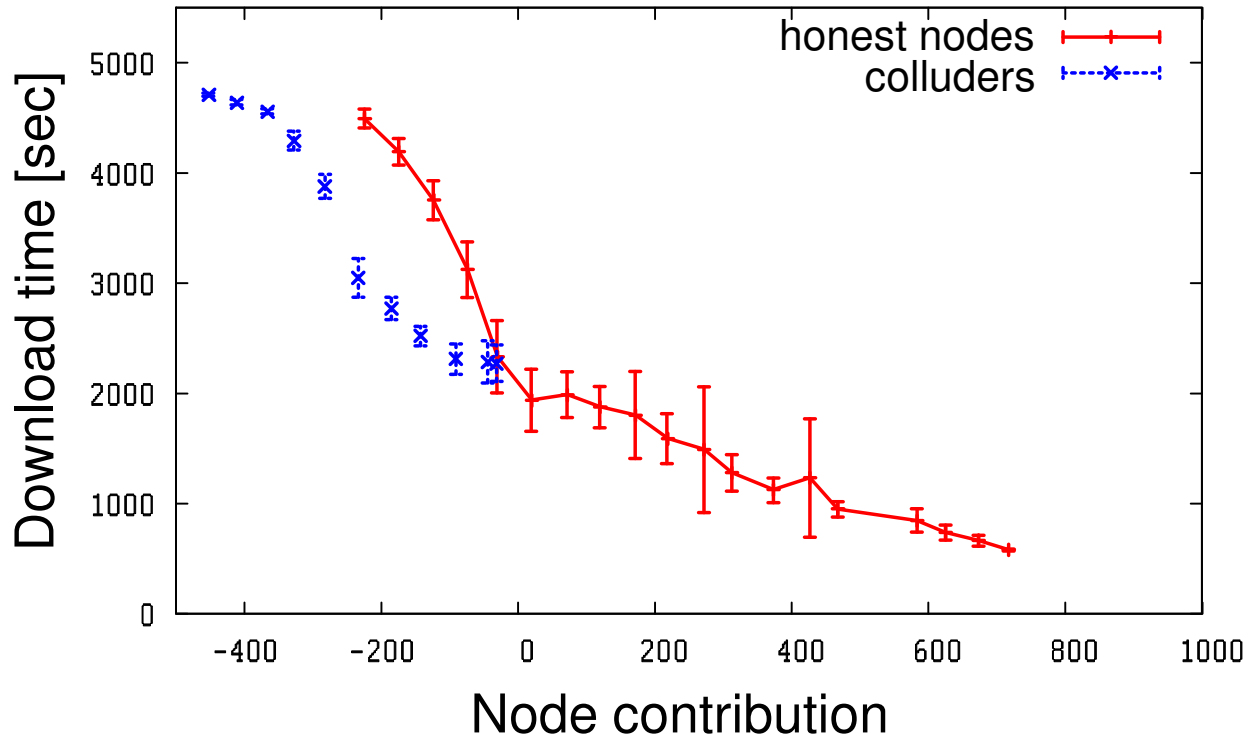
Higher reputation \rightarrow faster download



Higher reputation → faster download

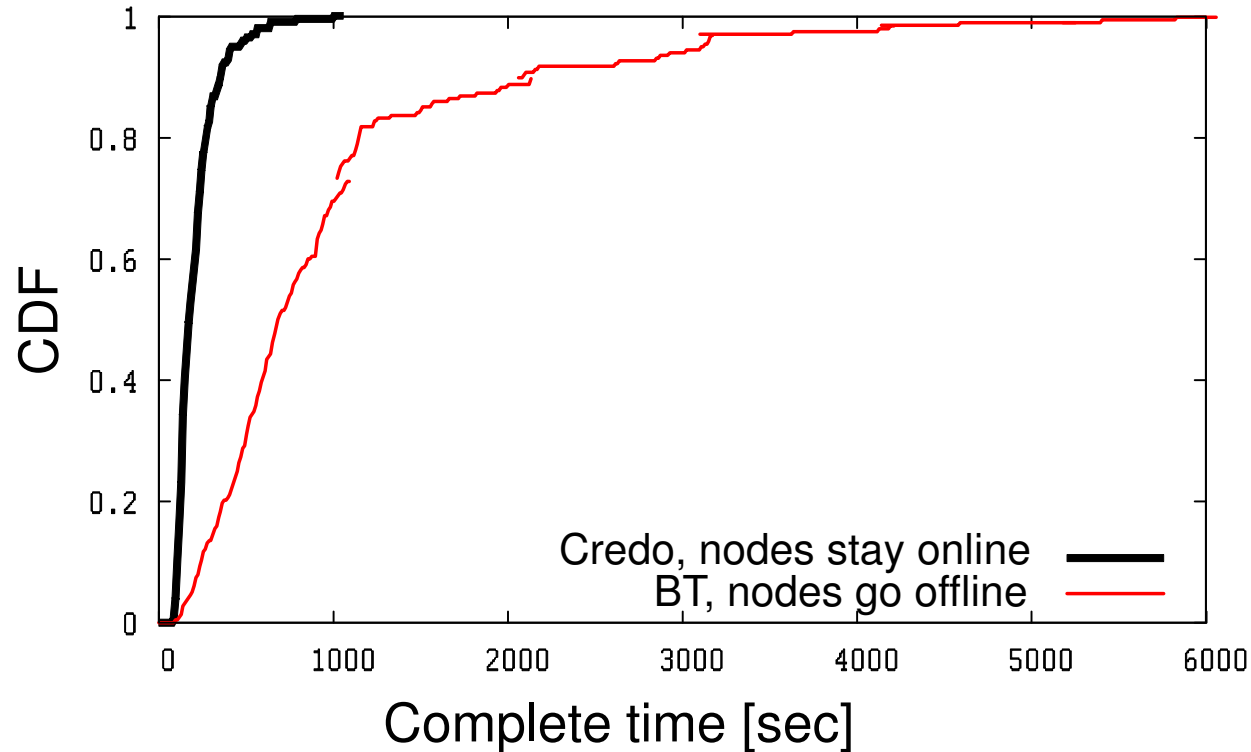


Credo is robust against collusion



- 30 adversaries, each brings in 3 Sybil nodes
- Colluders do not upload
- Vary demand of colluders at each run of the simulation

More seeders → better performance



- Experiment on 210 PlanetLab nodes
- Inject 25MB file at the beginning
- Nodes arrives every 15 second

Related work

- Graph-based reputation
 - Page-rank style: EigentTrust [WWW'03], multi-level tit-for-tat [IPTPS'06]
 - Max-flow style: SybilProof [P2PEcon'05], Feldman [EC'04]
 - Other: Onehop [NSDI'09]
- Currency
 - Dandelion [Usenix'07], Pace [Conext'08], Ppay [CCS'03]

Conclusion

- Credo addresses seeder promotion problem
 - Higher reputation → faster download
- Credo is a credit-based reputation system
 - Reflect nodes' net contribution correctly
 - Resilient to Sybil and collusion attacks