ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

Aula 15: Coordenação (parte 4)

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

22/04/2024

Na aula passada...

- Algoritmos de exclusão mútua
- Algoritmos de eleição de líder
  - por intimidação
  - em anel

Exemplo: Eleição de líder em grupo de servidores ZooKeeper

Noções básicas

- Cada servidor s no grupo de servidores tem um identificador $id(s)$
- Cada servidor tem um contador monotonamente crescente $tx(s)$ da última transação que ele tratou.
- Quando o seguidor s suspeita que o líder falhou, ele transmite uma mensagem de *ELECTION*, junto com o par ($voteID$,$voteTX$). Inicialmente,
  - $voteID \leftarrow id(s)$
  - $voteTX \leftarrow tx(s)$
- Cada servidor s mantém duas variáveis:
  - $leader(s)$: registra o servidor que s acredita que pode ser o líder final. Inicialmente, $leader(s) \leftarrow id(s)$.
  - $lastTX(s)$: transação mais recente conhecida por s. Inicialmente, $lastTX(s) \leftarrow tx(s)$.

Exemplo: Eleição de líder em grupo de servidores ZooKeeper

Quando s^* recebe ($voteID$,$voteTX$)

- Se $lastTX(s^*) < voteTX$, s^* acabou de receber informações mais atualizadas sobre a transação mais recente. Então faz:
  - $leader(s^*) \leftarrow voteID$
  - $lastTX(s^*) \leftarrow voteTX$
- Se $lastTX(s^*) = voteTX$ *E* $leader(s^*) < voteID$, então s^* tem a mesma informação sobre a transação mais recente, igual a que acabou de receber. Porém, é preciso atualizar qual servidor será o próximo líder (baseado no ID):
  - $leader(s^*) \leftarrow voteID$

Nota

Quando s^* acredita que deve ser o líder, ele transmite $\langle id(s^*), tx(s^*)\rangle$.
Ou seja, é basicamente intimidação.

Exemplo: Eleição de líder no Raft

Noções básicas

- Há um grupo (relativamente pequeno) de servidores
- Um servidor está em um dos três estados: *seguidor*, *candidato* ou *lider*
- O protocolo funciona em períodos (*terms*), começando com o período 0
- Cada servidor começa no estado *seguidor*.
- Um líder deve transmitir mensagens regularmente (talvez apenas um simples heartbeat)

Exemplo: Eleição de líder no Raft

Selecionando um novo líder

Após o seguidor s^* não receber nada do suposto líder s por algum tempo, s^* transmite uma mensagem se voluntariando para ser o próximo líder, aumentando o período em 1. s^* vai para o estado candidato. Então:

- Se o líder s receber a mensagem, ele responde confirmando que ainda é o líder. s^* retorna ao estado seguidor.
- Se outro seguidor s^{**} receber a mensagem de eleição de s^* e for a primeira mensagem de eleição durante o termo atual, s^{**} vota em s^*. Caso contrário, ele simplesmente ignora a mensagem de eleição de s^*. Quando s^* coleta a maioria dos votos, um novo período começa com um novo líder.

Exemplo: Eleição de líder no Raft

Selecionando um novo líder

Após o seguidor s^* não receber nada do suposto líder s por algum tempo, s^* transmite uma mensagem se voluntariando para ser o próximo líder, aumentando o período em 1. s^* vai para o estado candidato. Então:

- Se o líder s receber a mensagem, ele responde confirmando que ainda é o líder. s^* retorna ao estado seguidor.
- Se outro seguidor s^{**} receber a mensagem de eleição de s^* e for a primeira mensagem de eleição durante o termo atual, s^{**} vota em s^*. Caso contrário, ele simplesmente ignora a mensagem de eleição de s^*. Quando s^* coleta a maioria dos votos, um novo período começa com um novo líder.

Observação

Ao adicionar pequenas diferenças nos valores de timeout para decidir quando iniciar uma eleição de cada seguidor, podemos evitar eleições simultâneas, e a eleição convergirá rapidamente.

Eleições por prova de trabalho (Proof of Work)

Noções básicas

- Considere um grupo potencialmente grande de processos
- Cada processo é obrigado a resolver um desafio computacional
- Quando um processo resolve o desafio computacional, ele transmite sua vitória para o grupo
- Supomos que haja um procedimento de resolução de conflitos quando mais de um processo reivindica a vitória

Eleições por prova de trabalho (Proof of Work)

Resolvendo um desafio computacional

- Escolha uma função de hash segura $H(m)$:
  - Dada uma cadeia de bits m qualquer; $H(m)$ retorna uma cadeia de bits de comprimento fixo
  - calcular $h = H(m)$ é computacionalmente eficiente
  - encontrar uma função H^{-1} tal que $m = H^{-1}(H(m))$ é computacionalmente extremamente difícil
- Na prática: encontrar H^{-1} se resume a um extenso procedimento de tentativa e erro

Eleições por prova de trabalho

Corrida controlada

- Suponha uma função de hash segura globalmente conhecida H^*.
- Bloco de transações m
- Tarefa: dada uma cadeia de bits $h = H^*(m)$, encontrar uma cadeia de bits $\tilde{h}$ tal que $h^* = H^*(\tilde{h} \odot h)$ onde:
  - h^* deve ser uma cadeia de bits com K zeros à esquerda
  - $\tilde{h} \odot h$ denota alguma operação a nível de bit predeterminada feita com $\tilde{h}$ e h

Eleições por prova de trabalho

Observação

Controlando K, controlamos a dificuldade de encontrar $\tilde{h}$. Se p é a probabilidade de que um palpite aleatório para $\tilde{h}$ seja correto: $p = (1/2)^K$.

Eleições por prova de trabalho

Observação

Controlando K, controlamos a dificuldade de encontrar $\tilde{h}$. Se p é a probabilidade de que um palpite aleatório para $\tilde{h}$ seja correto: $p = (1/2)^K$.

Prática atual

Em muitos sistemas de blockchain baseados em PoW, $K = 64$

- Com $K = 64$, leva cerca de 10 minutos em um supercomputador para encontrar $\tilde{h}$
- Com 2500 transações por bloco: 4 transações/s
- Com $K = 64$, leva cerca de 100 anos em um laptop para encontrar $\tilde{h}$

Eleições por prova de participação (proof of stake)

Hipóteses

Assumimos um sistema de blockchain no qual N tokens seguros são utilizados:

- Cada token tem um proprietário único
- Cada token tem um índice exclusivamente associado $1 \leq k \leq N$
- Um token não pode ser modificado ou copiado sem que isso passe despercebido

Princípio

- Sortear um número aleatório $k \in \{1, \ldots, N\}$
- Procurar o processo P que possui o token com índice k. P é o próximo líder.

Observação

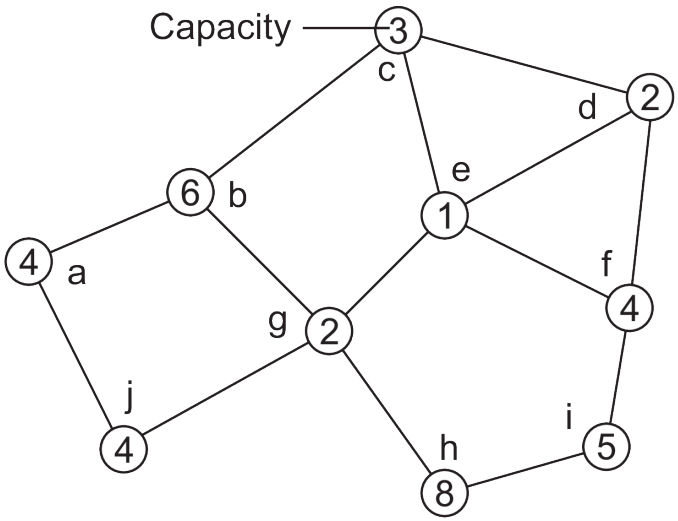
Quanto mais tokens um processo possui, maior a probabilidade de ser selecionado como líder.

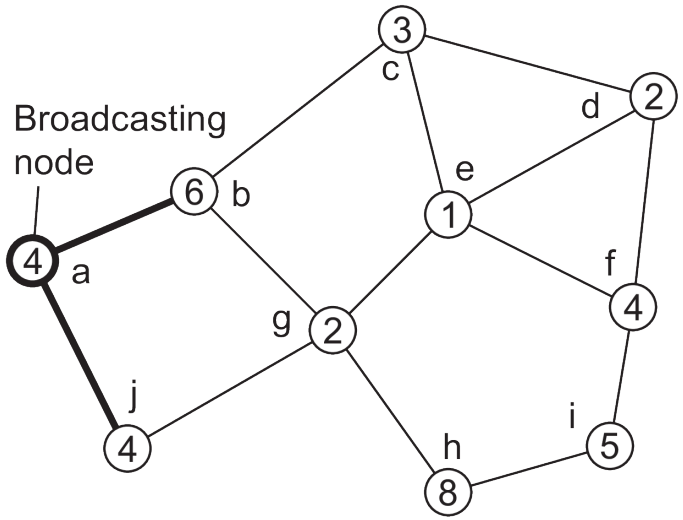
Eleições em ambientes sem fio

- Transmissão pouco confiável
- Mudanças de topologia
- Possibilidade de particionamento da rede

Uma solução para redes sem fio

Um exemplo de rede



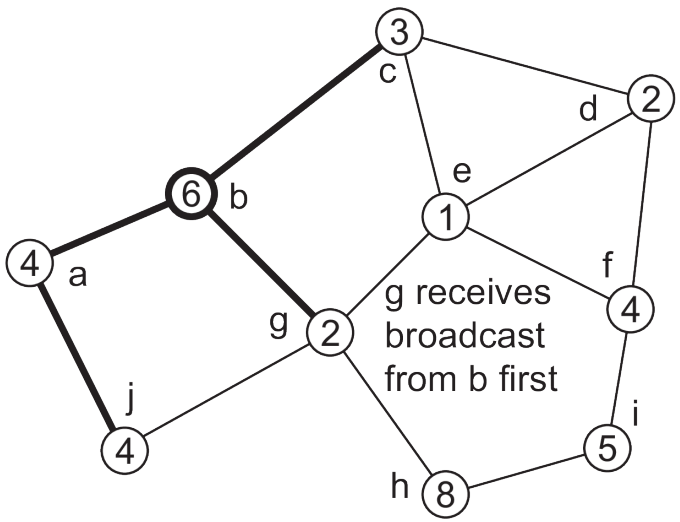


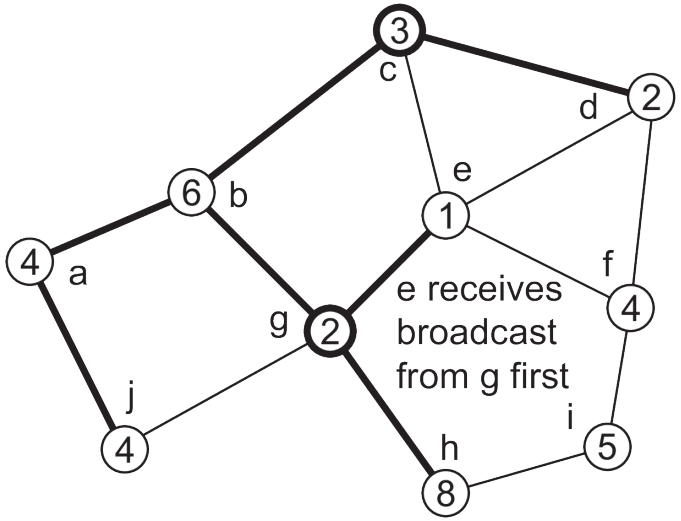
Essência

Encontrar o nó com a capacidade mais alta para ser selecionado como próximo líder.

Uma solução para redes sem fio

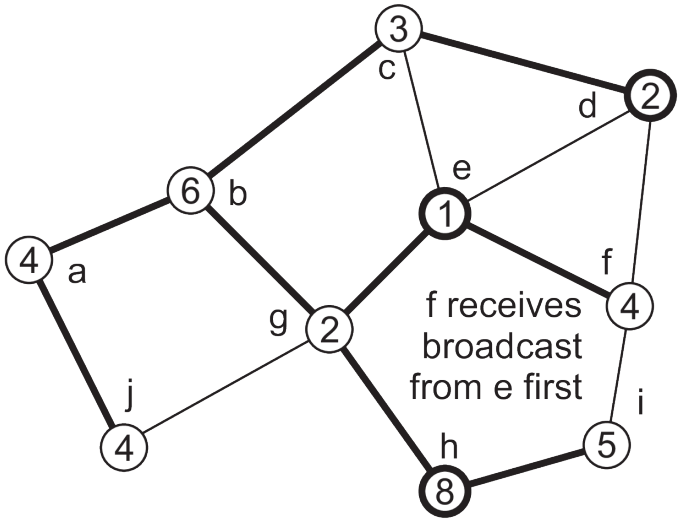
Um exemplo de rede

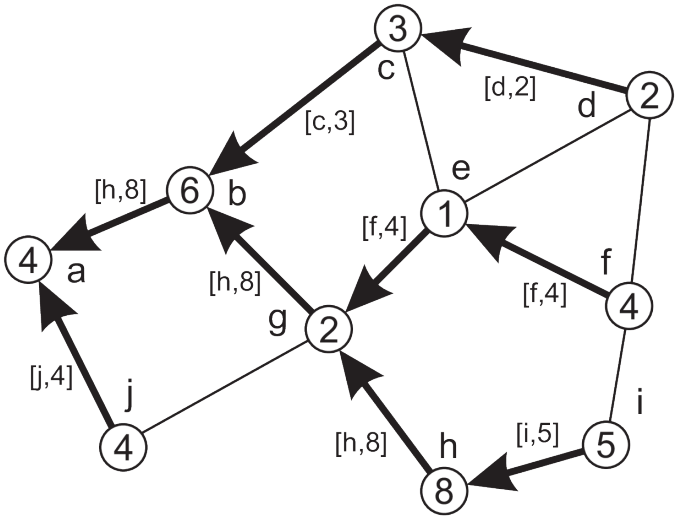




Uma solução para redes sem fio

Um exemplo de rede





Essência

Um nó reporta apenas o nó que descobriu ter a maior capacidade.

Coordenação baseada em gossip: agregação

Aplicações típicas

- Disseminação de dados: Talvez o mais importante. Note que existem muitas variantes de disseminação.
- Agregação: cada nó P_i mantém uma variável v_i. Quando dois nós trocam mensagens, cada um redefine sua variável para

$$v_i, v_j \leftarrow (v_i + v_j)/2$$

Resultado: no final cada nó terá computado a média $\bar{v} = \sum_i v_i/N$.

Coordenação baseada em gossip: agregação

Aplicações típicas

- Disseminação de dados: Talvez o mais importante. Note que existem muitas variantes de disseminação.
- Agregação: cada nó P_i mantém uma variável v_i. Quando dois nós trocam mensagens, cada um redefine sua variável para

$$v_i, v_j \leftarrow (v_i + v_j)/2$$

  Resultado: no final cada nó terá computado a média $\bar{v} = \sum_i v_i/N$.
- O que acontece no caso em que inicialmente $v_i = 1$ e $v_j = 0, j \neq i$?

Observação

Outras funções podem ser utilizadas, e.g. MAX, MIN.

Coordenação baseada em gossip: amostragem de pares

Problema

Para muitas aplicações baseadas em gossip, você precisa selecionar um peer uniformemente aleatório de toda a rede. Em princípio, isso significa que você precisa conhecer todos os outros peers. Impossível?

Noções básicas

- Cada nó mantém uma lista de c referências a outros nós
- Regularmente, escolha outro nó ao acaso (da lista) e troque aproximadamente $c/2$ referências
- Quando a aplicação precisa selecionar um nó aleatório, também escolhe um aleatório de sua lista local.

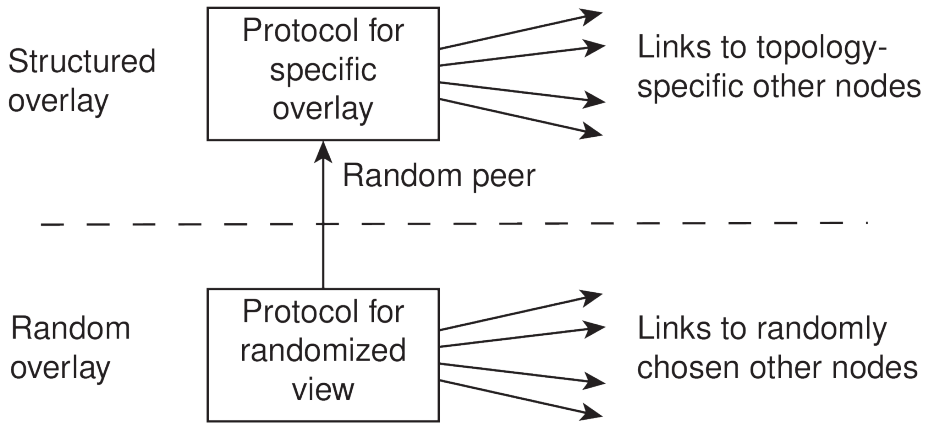
Observação

Estatisticamente, a seleção de um par da lista local é indistinguível da seleção aleatória uniforme de um par de toda a rede.

Construção de overlay baseada em gossip

Essência

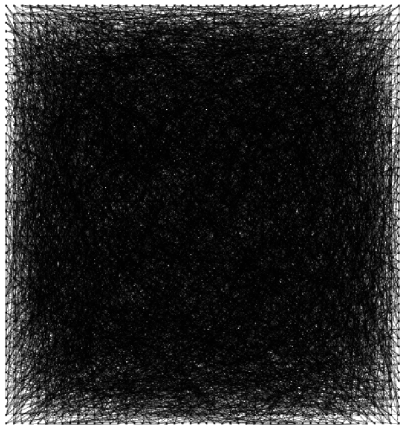
Mantenha duas listas locais de vizinhos. A de baixo é usada para fornecer um serviço de amostragem de pares; a lista de cima é usada para selecionar cuidadosamente vizinhos dependentes da aplicação.



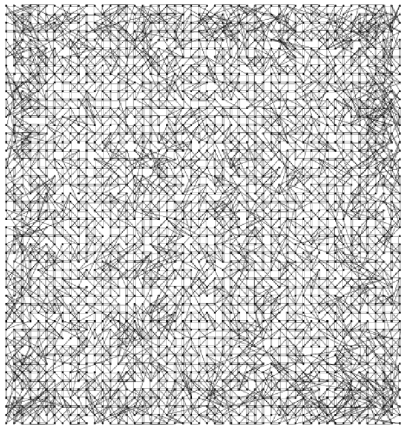
Construção de overlay baseada em gossip: um toro 2D

Considere um grid $N \times N$.

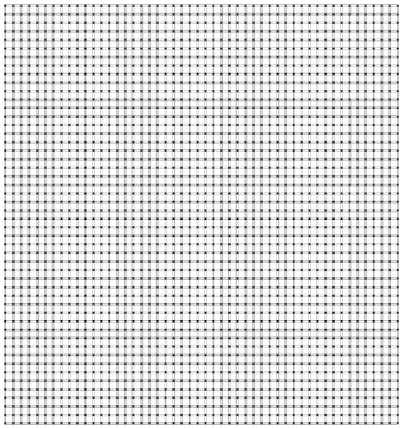
- Todo nó deve manter uma lista dos c vizinhos mais próximos
- Distância entre o nó em (a_1, a_2) e (b_1, b_2) é $d_1 + d_2$, com $d_i = \min(N - |a_i - b_i|, |a_i - b_i|)$
- Todo nó escolhe outro nó aleatório de sua lista de baixo e mantém apenas o mais próximo em sua lista de cima.
- Uma vez que todos os nós tenham escolhido e selecionado um nó aleatório, passamos para a próxima rodada.



início ($N = 50$)



após 5 rodadas

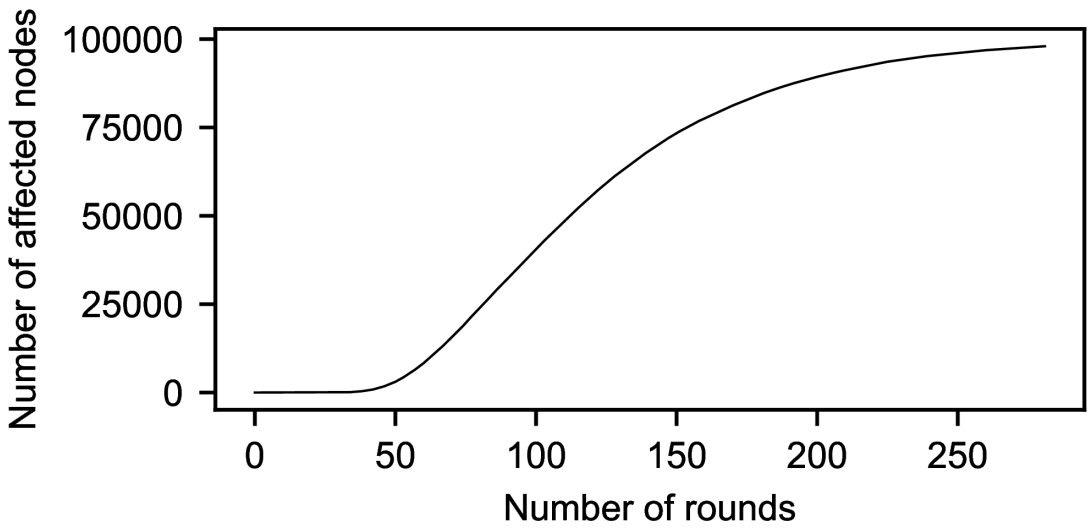


após 20 rodadas

Gossiping seguro

Ataque

Considere um conjunto de nós conspiradores que, ao trocar referências, sistematicamente retorna links apenas para outros nós conspiradores ⇒ estamos lidando com um ataque de hub.

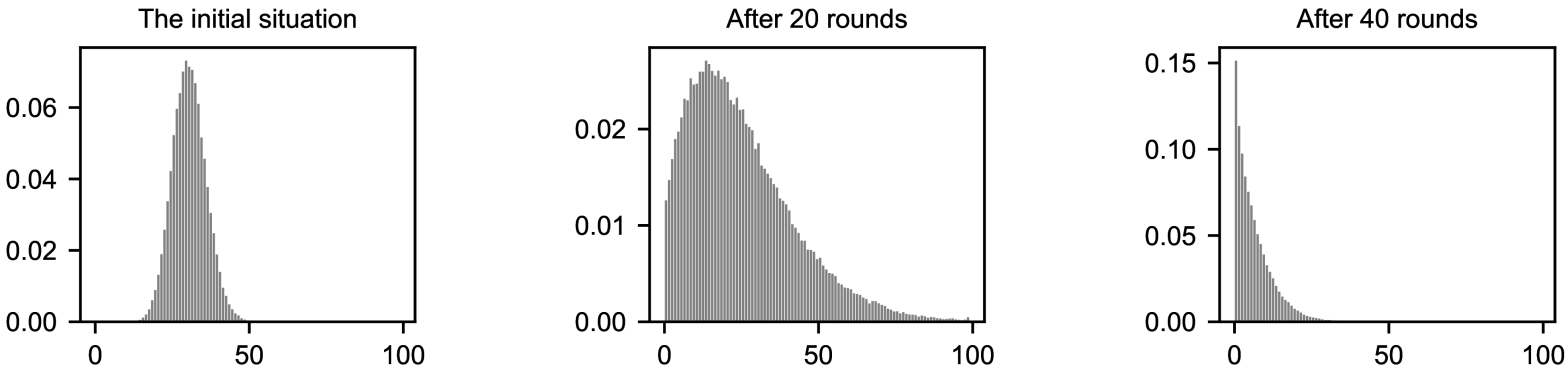


Situação

Uma rede com 100.000 nós, um tamanho de lista local $c = 30$, e apenas 30 atacantes. O eixo y mostra o número de nós com links apenas para os atacantes. Depois de menos de 300 rodadas, os atacantes têm controle total.

Uma solução: coleta de estatísticas

Medir as distribuições do grau de entrada entrada dos nós: qual fração de nós (eixo y) tem quantos outros nós apontando para eles (eixo x)?

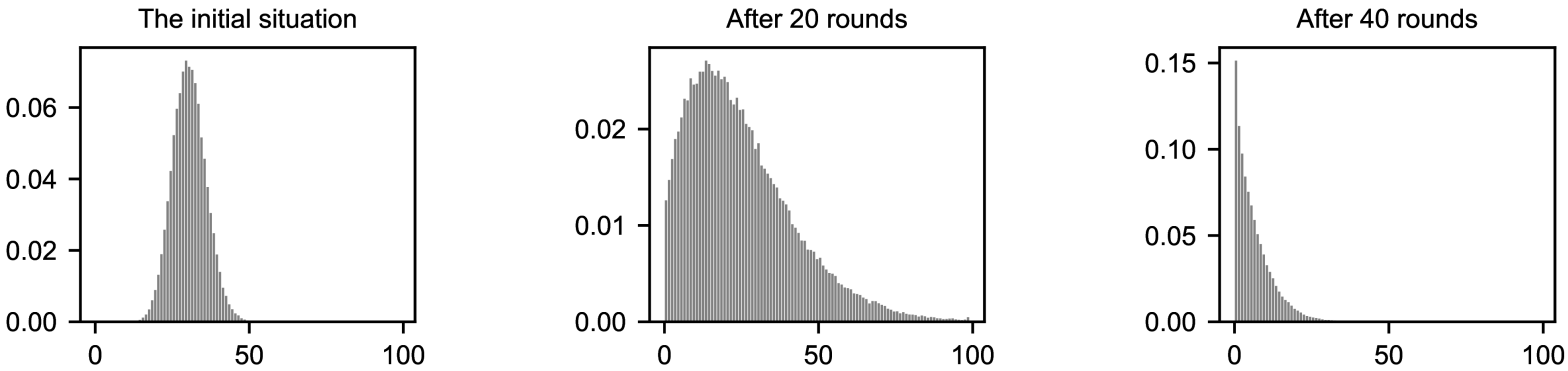


Abordagem básica

Quando um nó benigno inicia uma troca, ele pode usá-la para coletar estatísticas ou para atualizar sua lista local. O atacante fica no limbo: sua resposta será usada para fins estatísticos ou para fins atualizar a tabela?

Uma solução: coleta de estatísticas

Medir as distribuições do grau de entrada entrada dos nós: qual fração de nós (eixo y) tem quantos outros nós apontando para eles (eixo x)?



Abordagem básica

Quando um nó benigno inicia uma troca, ele pode usá-la para coletar estatísticas ou para atualizar sua lista local. O atacante fica no limbo: sua resposta será usada para fins estatísticos ou para fins atualizar a tabela?

Observação

No caso em que a coleta de estatísticas pode revelar conspiradores, um nó conspirador será forçado a se comportar de acordo com o protocolo.