Clouding Computing Final Presentation

Team King Crab: Xujia Cao, Yanxi Chen, Shaoyi Zhang

Presentation Outline

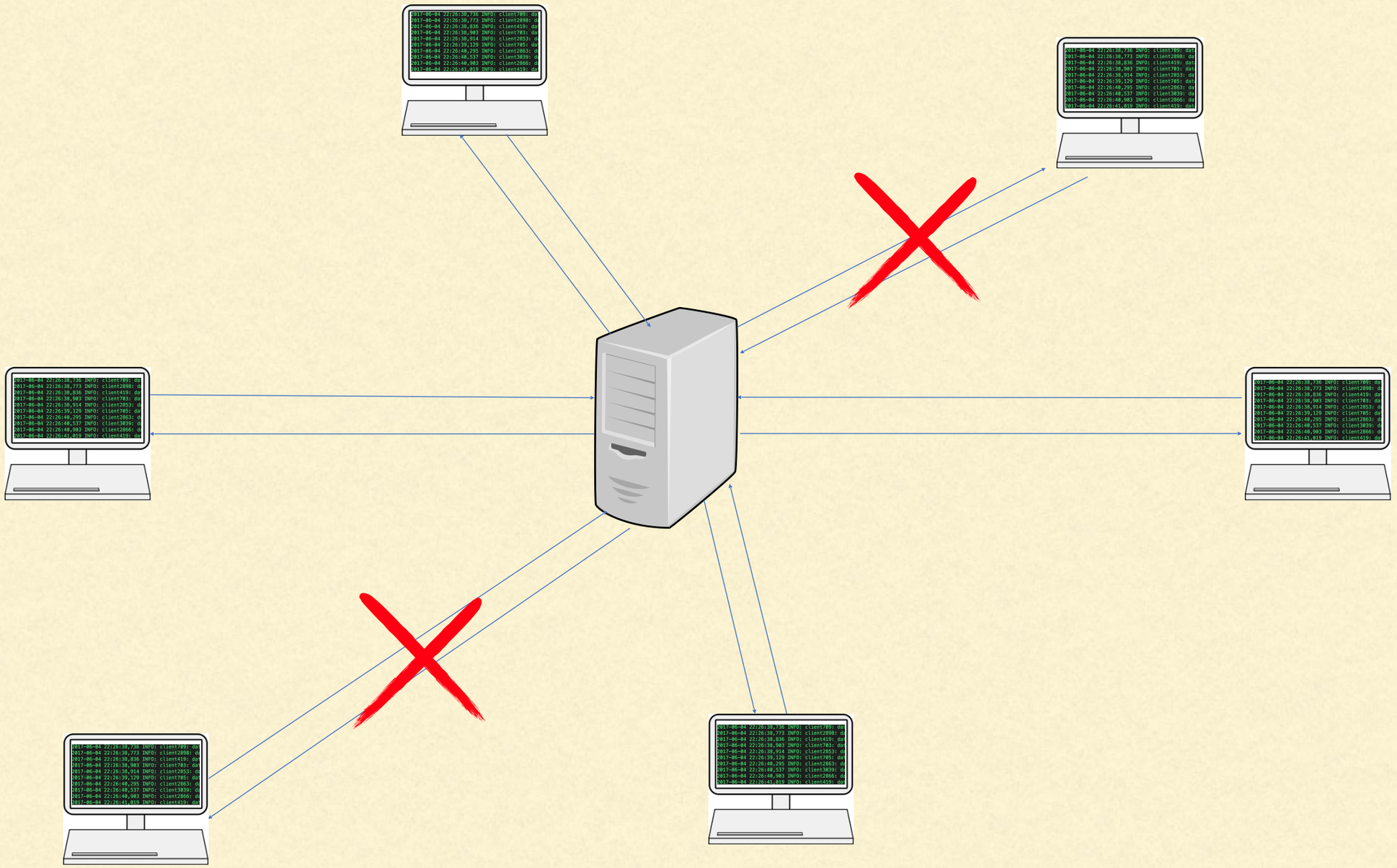
Computation Model

What doesn't work

What does work

System Design and Others

Computation Model



What doesn't work — things that we tried but not included in final implementation

Simulated Annealing

Stochastic Tunneling

Detrended Fluctuation Analysis

Clique Counts Approximation

Simulated Annealing/Stochastic Tunneling

$$e^{-\frac{\Delta f}{kT}} \geq \mathbb{R}(0, 1)$$

$$f_{\mathrm{STUN}}(\vec{x}) = 1.0 - e^{-\frac{\mathbb{E}(\vec{x}) - \mathbb{E}(\vec{x_0})}{\gamma}}$$

Challenges

- Reset parameters
- Too slow!!!!

Detrended Fluctuation Analysis (DFA)

Improving FPGA Placement with
Dynamically Adaptive Stochastic Tunneling

Clique Counts Approximation

A Fast and Provable Method for Estimating Clique Counts Using Turán's Theorem

Things that works

Tabu Search*

- Use “Tabu Search” to avoid walking to the same graph twice
- Tabu Search uses linked-list to store history
- However, we use a global Bloom Filter to replace linked-list

Global Bloom Filter (GBF)

- Client ask GBF whether graph G has been checked before
- Case 1:
  - If G is indeed checked before, GBF will tell us G has been reached with probability 1
- Case 2:
  - If G hasn't been checked before, GBF might give a False Positive response (yeah, G has been checked) with probability p

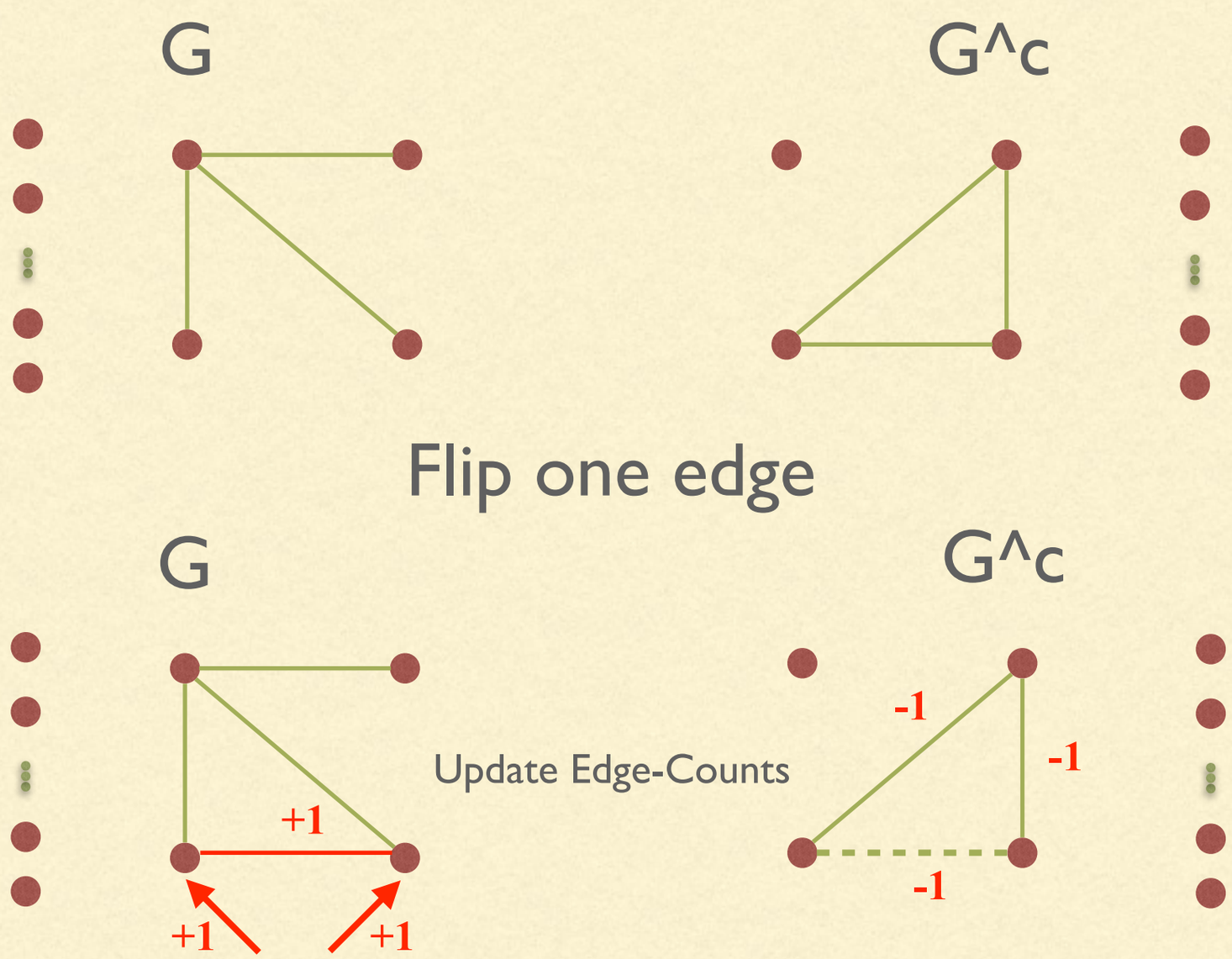
Global Bloom Filter (GBF)

- Benefits of GBF
  - Space efficient because:
    - only store bits (similar to hash) instead of the giant graph/matrix representation
  - Parametrizable because:
    - Configure possible solution number N and false positive rate P
    - For example,
      - N = 40,000,000, p =1.0E-10: size = 228.53MB

The main trick

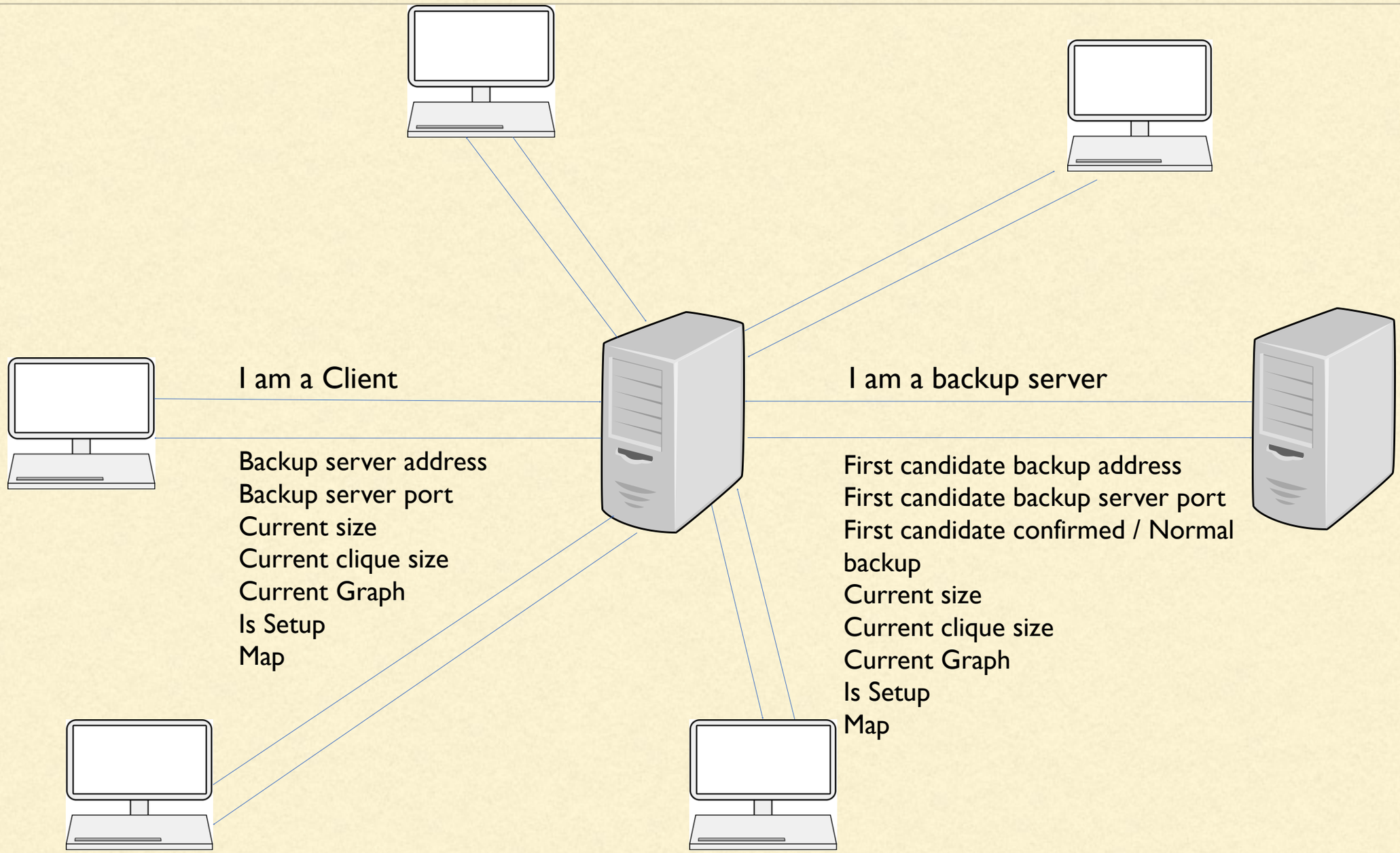
- When move to larger problem size
  - Count 10 cliques once
  - Maintain a Map: {Edges} -> {clique counts}
  - After the first and only10-clique count check, we start to flip edges
    - During flipping, We also maintain the "complement" of G, because:
      - To check R(m,n) is the same to check:
        1. m-clique on G
        2. n-clique on G^c

G's complement and (n-2)-clique

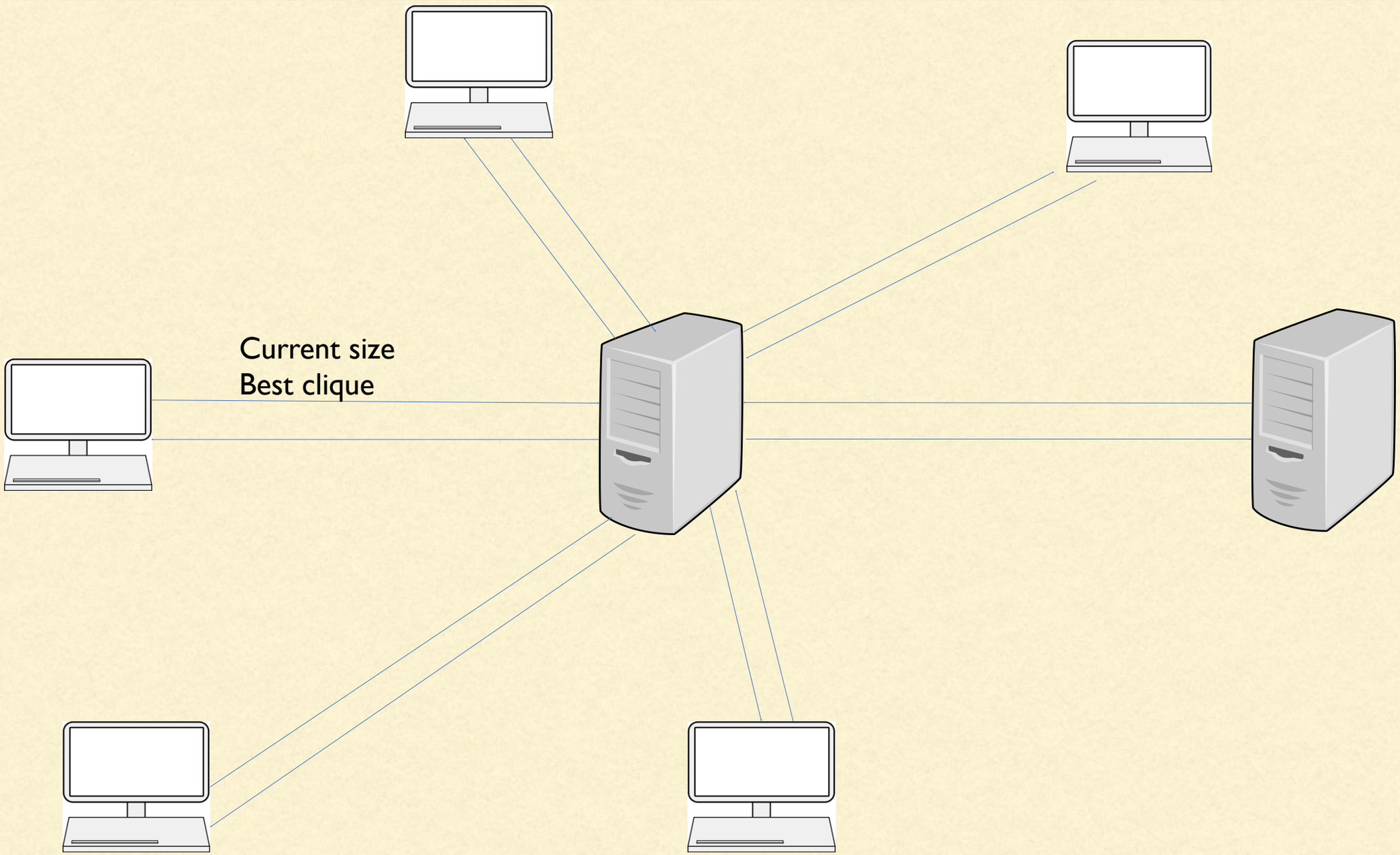


Search for (n-2)-clique in the set of intersected neighbors

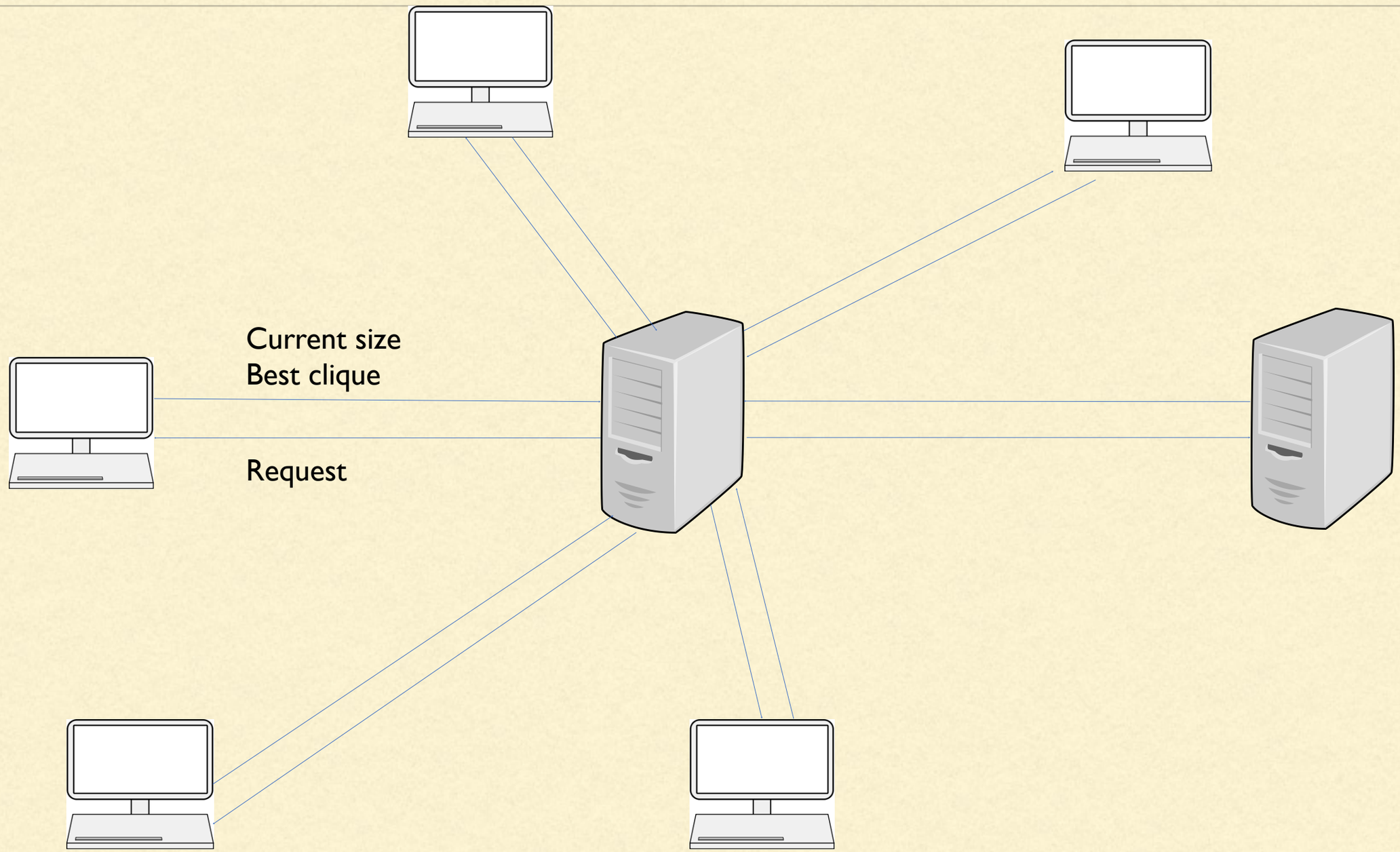
Server-Client Protocol



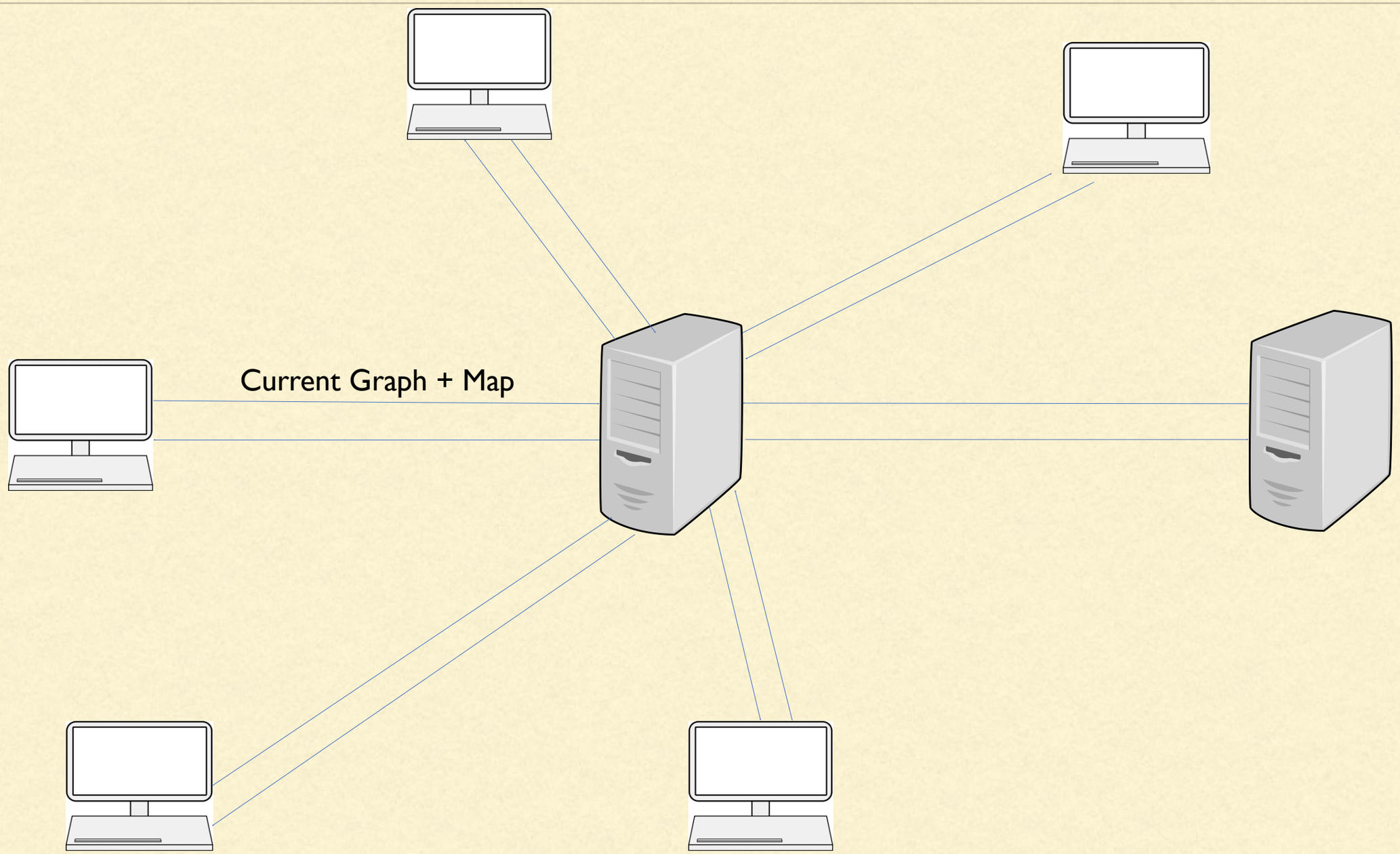
Server-Client Protocol



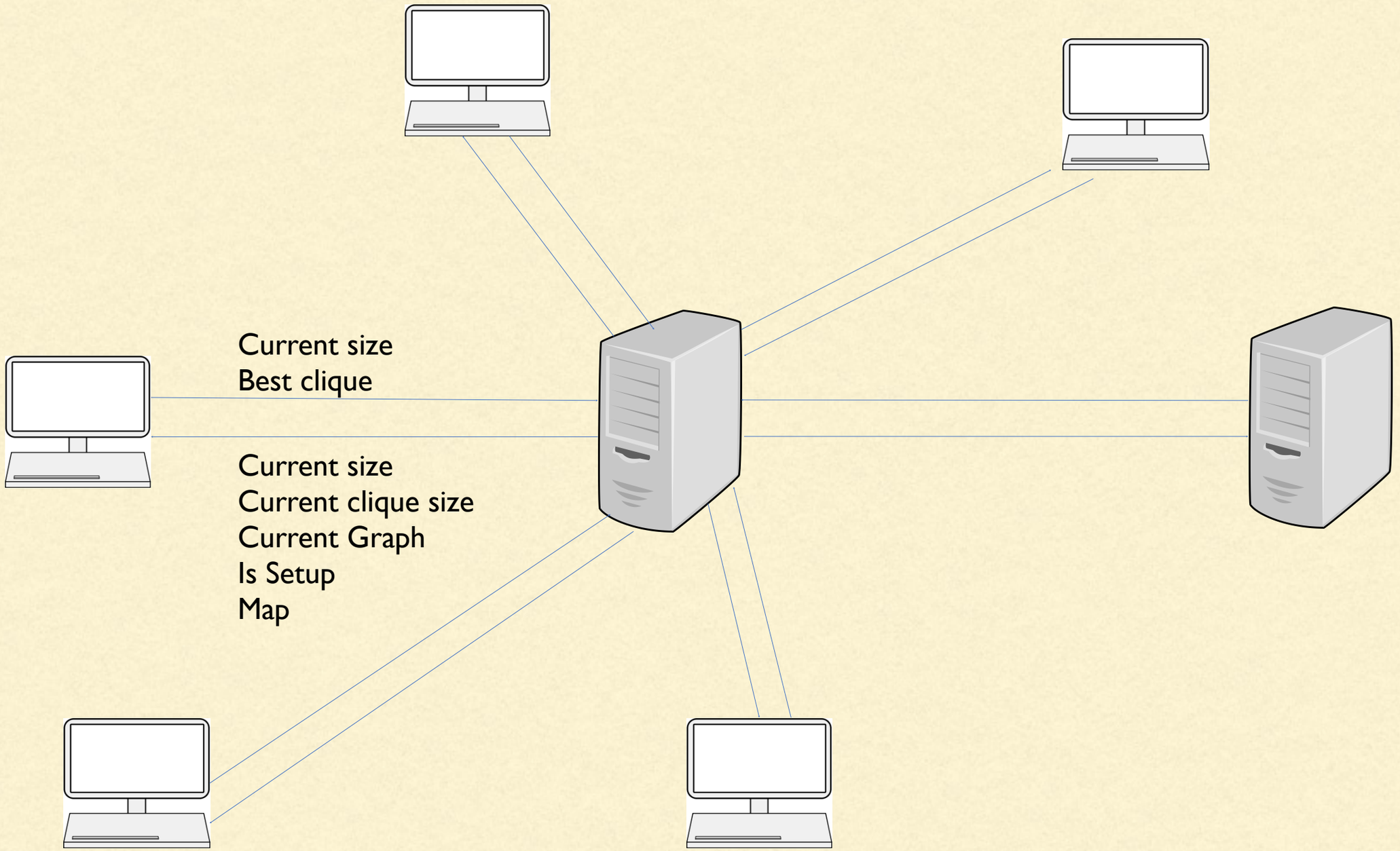
Server-Client Protocol



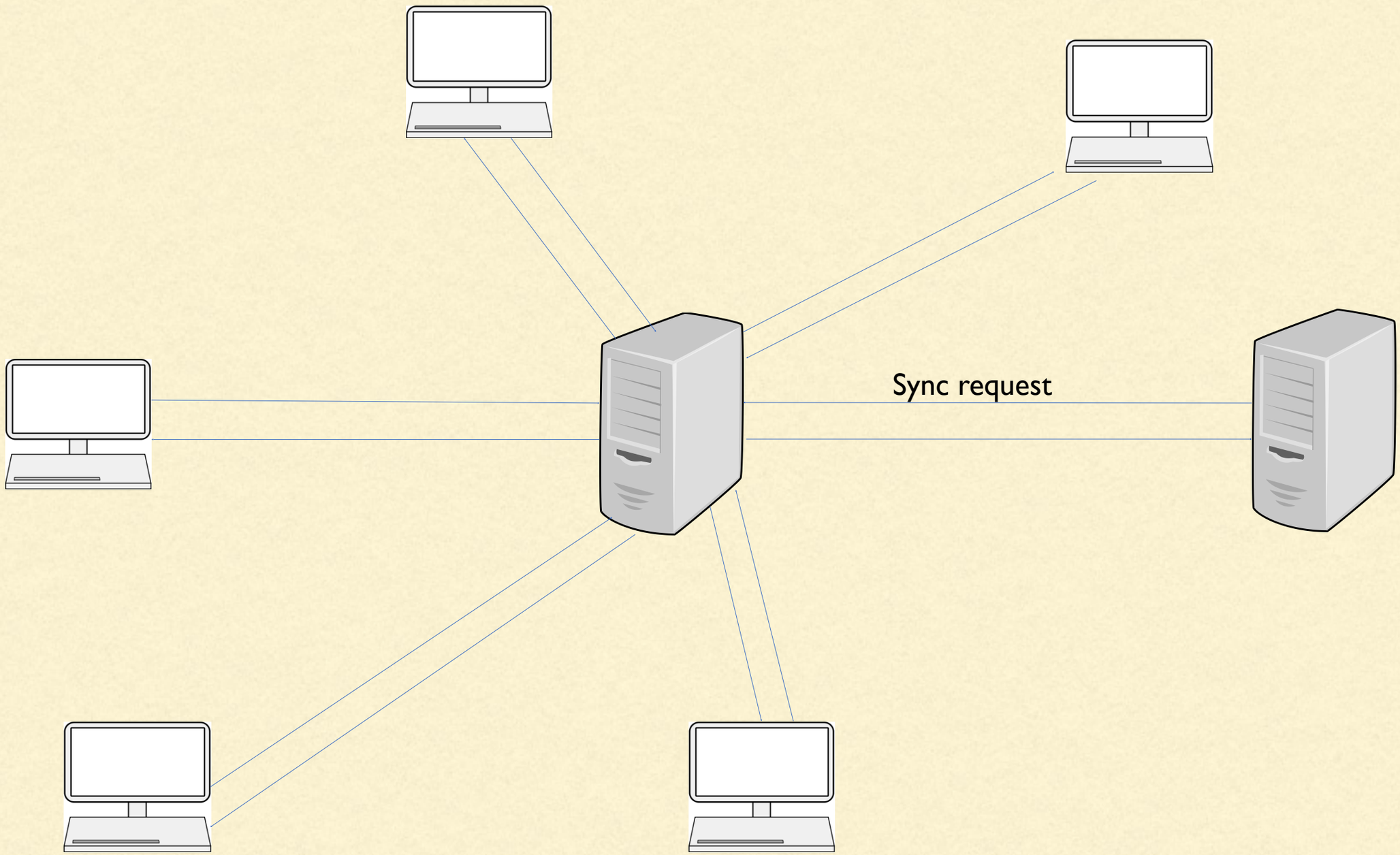
Server-Client Protocol



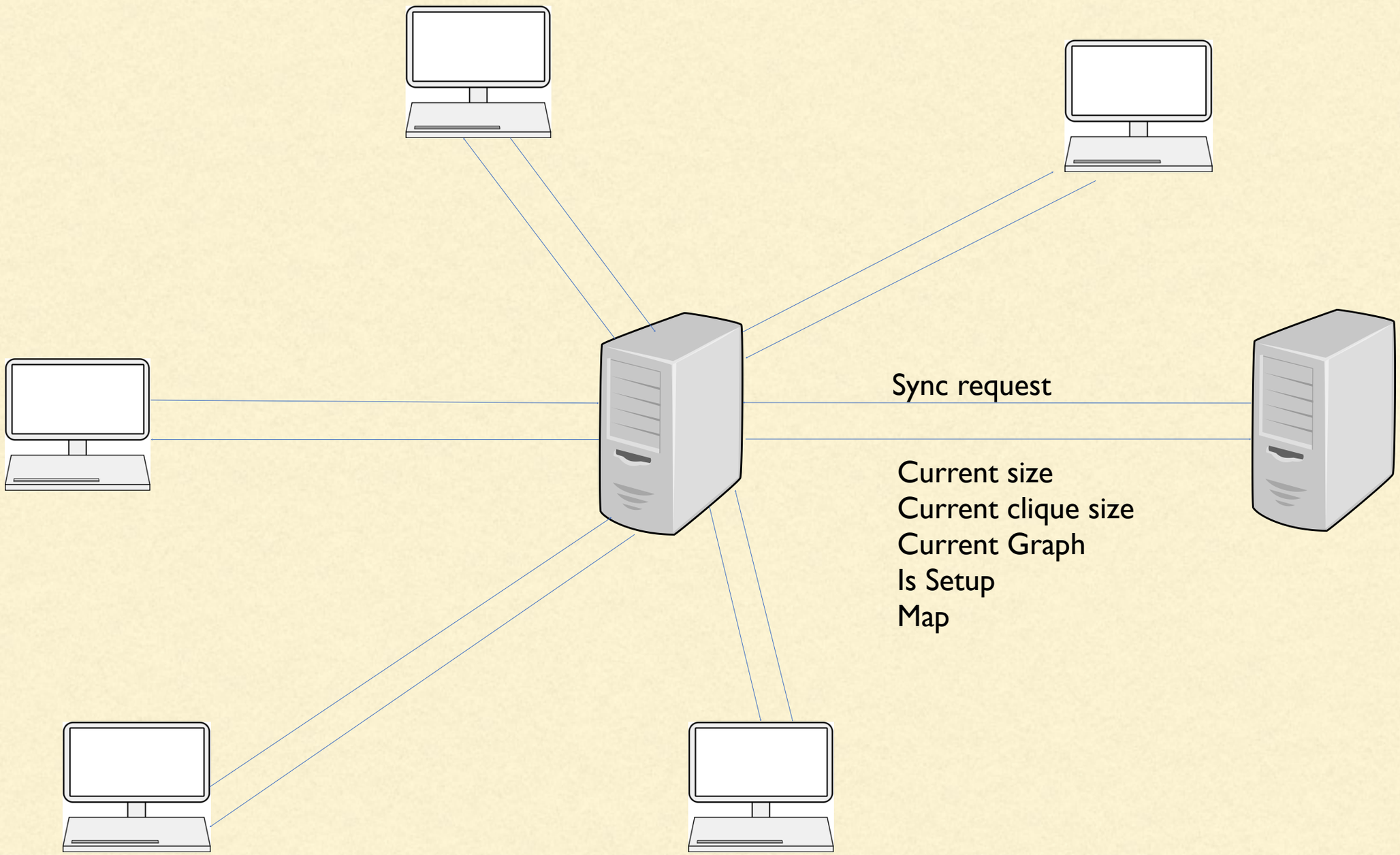
Server-Client Protocol



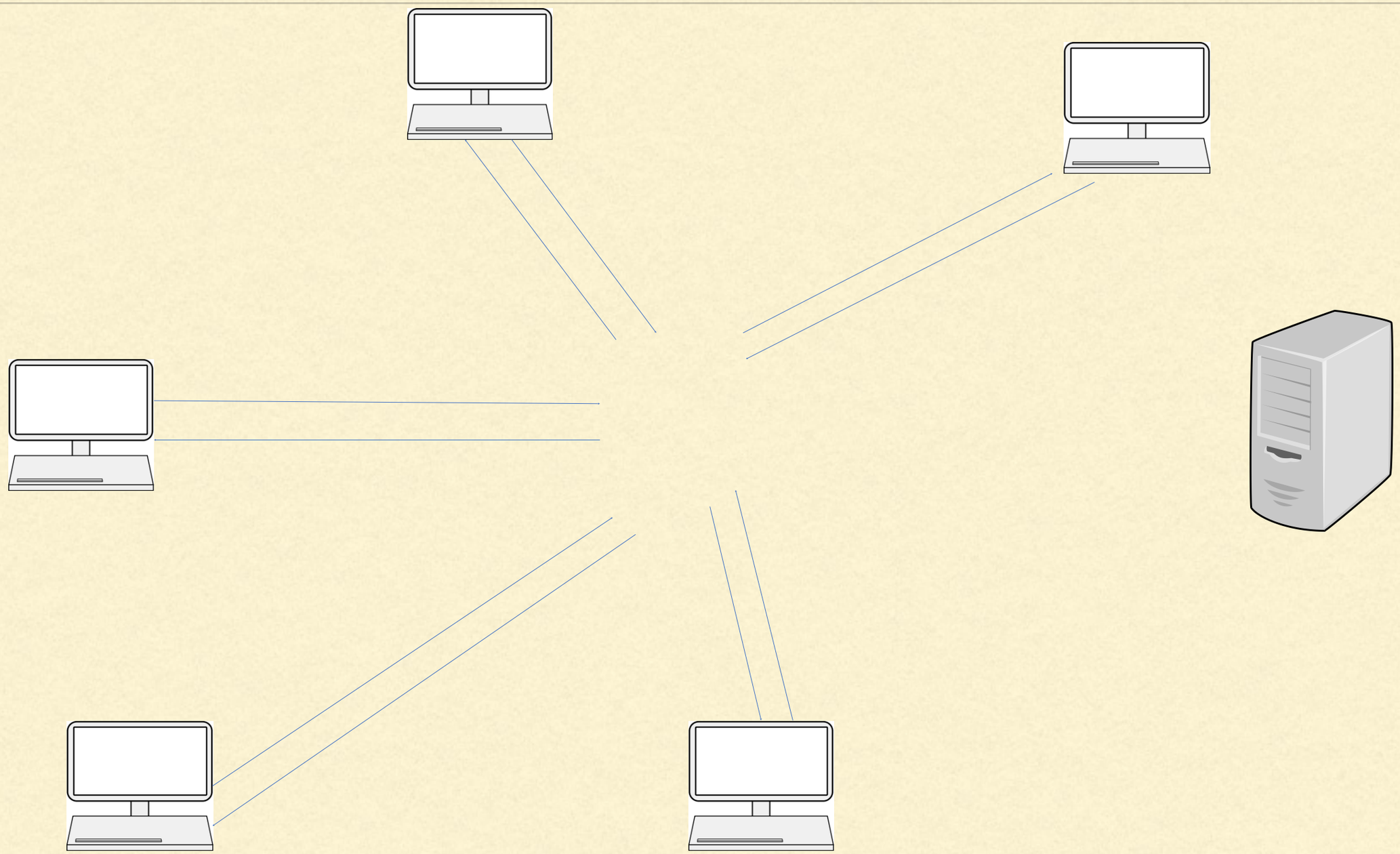
Server-Client Protocol



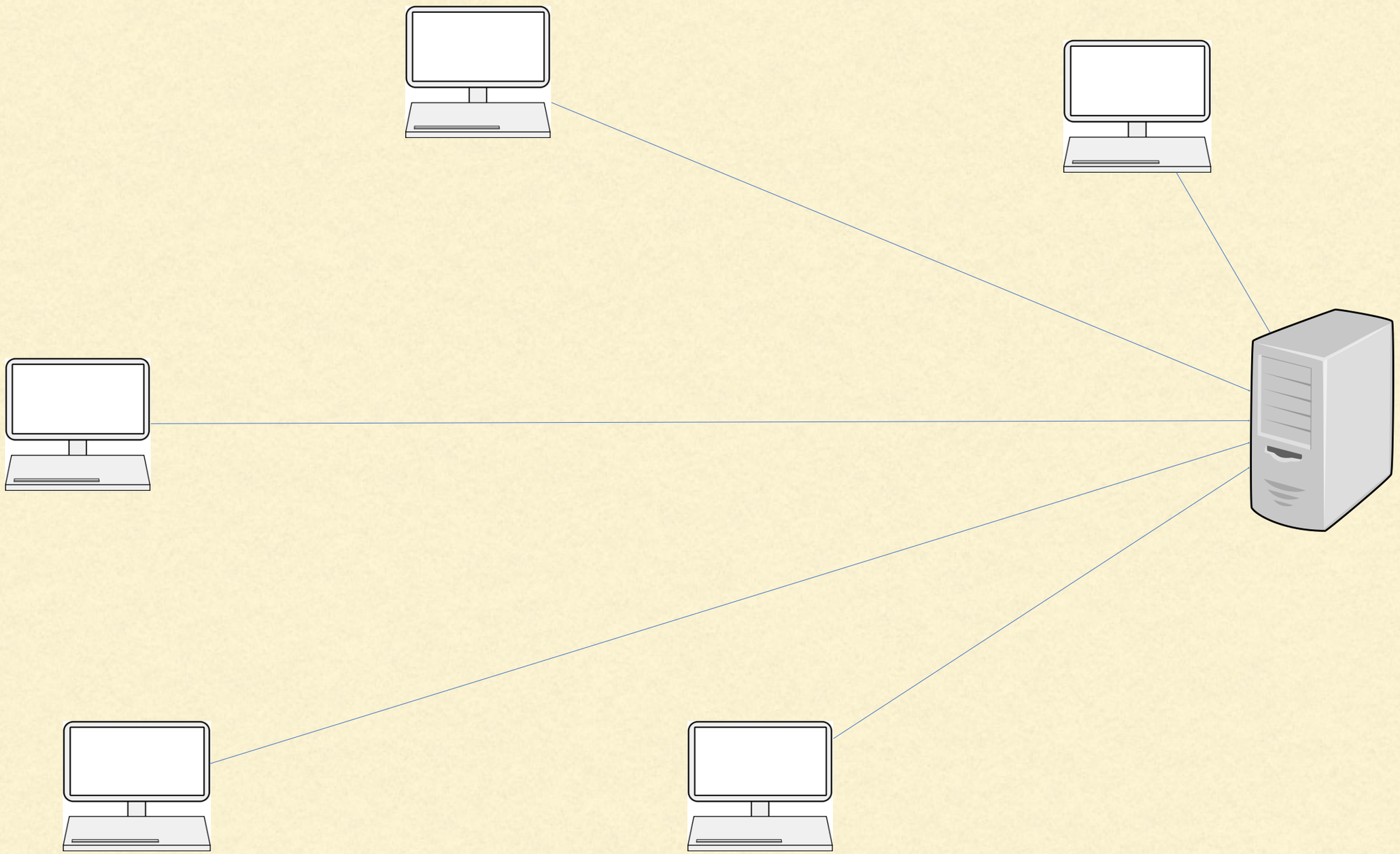
Server-Client Protocol



Server-Client Protocol



Server-Client Protocol



Results

Best Clique Size:
340

CPU counts:

885842(iteration per day) * 7*10^6(CPU per iteration) * 35(days)

=2.14*10^14

Takeaways

Start early and do more logging