Distributed Data System by Random Network Coding

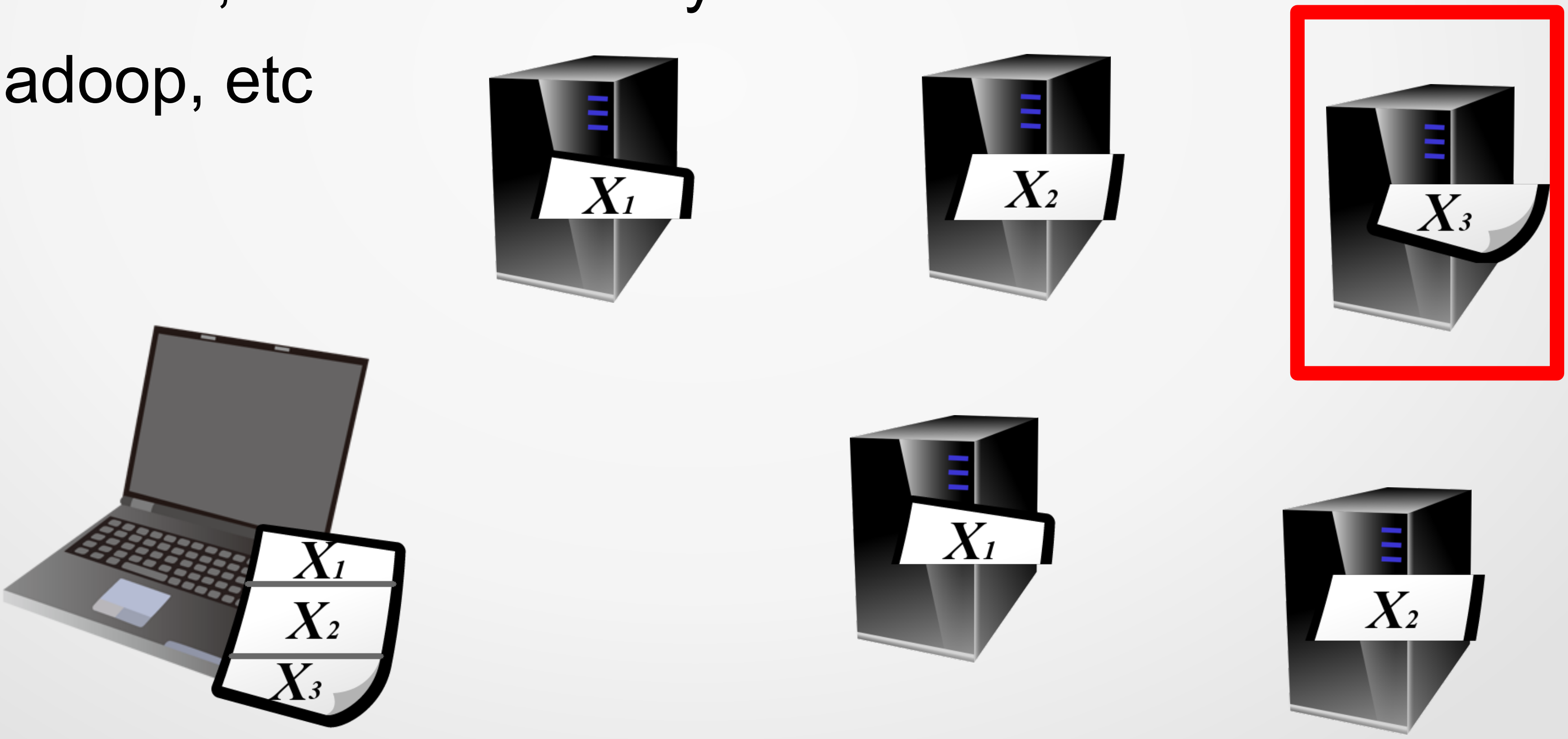
ASUSA Corporation
Hiroshi Nishida

General Distributed Storage System

- Traditional Distributed Storage System
  - All servers have the same raw files
  - General approach for most Content Delivery Networks (CDN) like Netflix, Youtube

General Distributed Storage System

- Slightly Efficient System
  - To save overall disk space, files are split into pieces and they are sent to servers.
  - However, this is not always reliable
  - Hadoop, etc

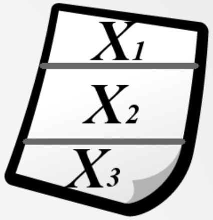


Distributed Storage System using Random Network Coding

- More Advanced System by RNC or Erasure Coding
  - Saves disk space and is more reliable
  - Any combination of two servers can fail
  - Each server stores only 1/3 of the original file size.

Distributed Storage System using Random Network Coding

- Principle of Random Network Coding
  - Split a file into three pieces – X_1, X_2, X_3
  - Randomly choose A_1, A_2, A_3, and calculate $B = A_1X_1 + A_2X_2 + A_3X_3$
  - Do it for B_1, B_2, …, $B_{\#\ of\ servers}$
  - For instance,



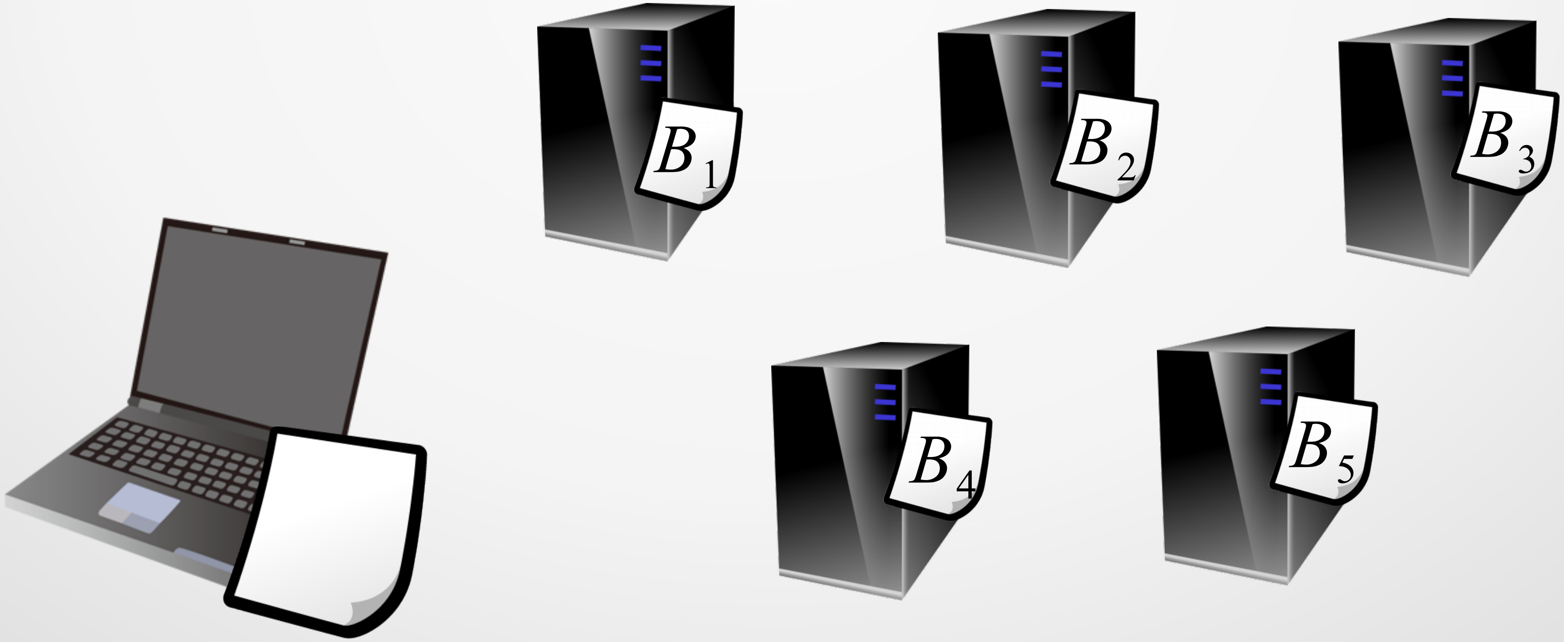
$$\begin{cases} B_1 & = & 3X_1 + 10X_2 + 7X_3 \\ B_2 & = & 8X_1 + 5X_2 + 2X_3 \\ B_3 & = & 1X_1 + 4X_2 + 23X_3 \\ B_4 & = & 11X_1 + 2X_2 + 9X_3 \\ B_5 & = & 4X_1 + 32X_2 + 11X_3 \end{cases}$$

Distributed Storage System using Random Network Coding

- Distribute B_1, B_2, …, B_5 to each server
- Note size of B_n (for all n) = size of X_k (for all k) = 1/3

  $$B = A_1 X_1 + A_2 X_2 + A_3 X_3$$

  because calculation is made in **Galois Field**
  (Actuall size of B_n is slightly greater than 1/3)



Distributed Storage System using Random Network Coding

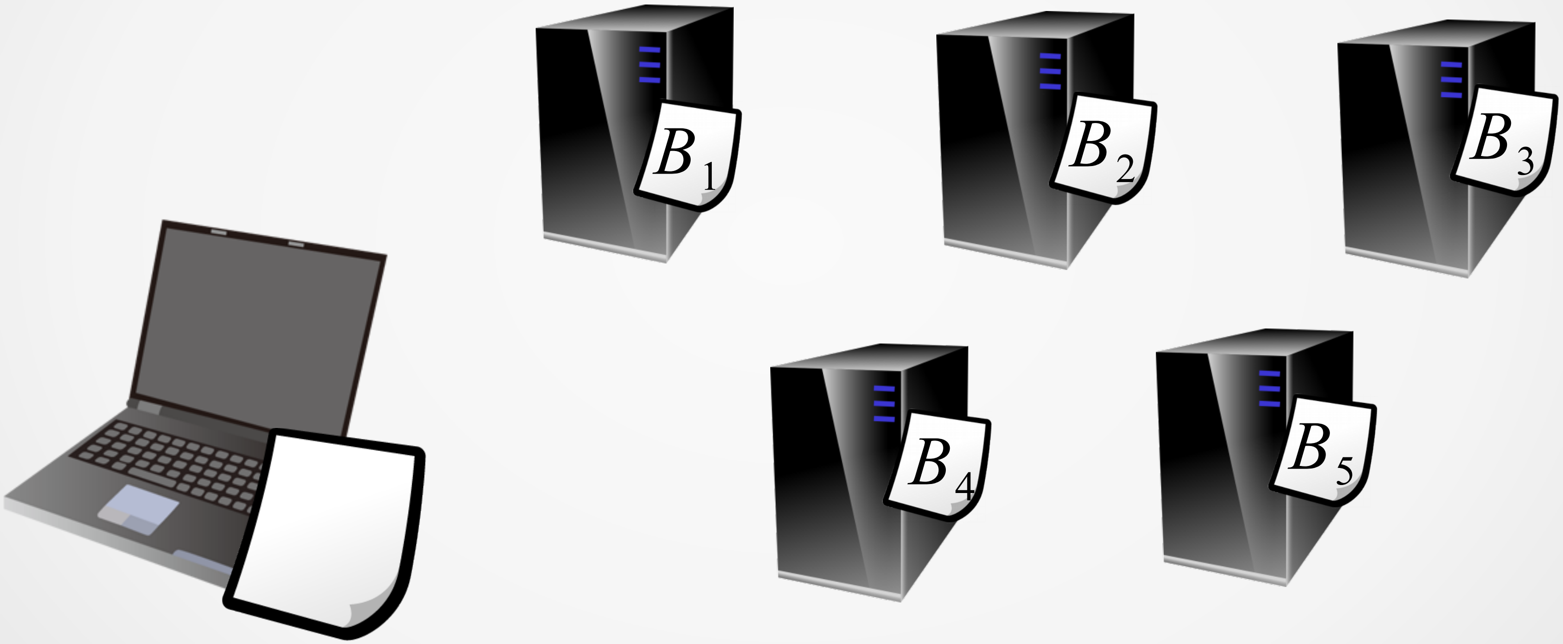
- Restoring Original File
  - With any three servers:

$$\begin{cases} B_2 & = & 8X_1+5X_2+2X_3 \\ B_3 & = & 1X_1+4X_2+23X_3 \\ B_5 & = & 4X_1+32X_2+11X_3 \end{cases}$$

  - We solve linear equations and obtain X_1, X_2, X_3
  - Concatenate

Distributed Storage System using Random Network Coding

- Saves disk space and achieves higher reliability



- Has high affinity to P2P

Distributed Storage System using Random Network Coding

- Eeasure Coding (RAID5, RAID6, etc...)

$$\begin{cases} B_1 & = & X_1 \\ B_2 & = & X_2 \\ B_3 & = & X_3 \\ B_4 & = & X_1 \oplus X_2 \oplus X_3 \end{cases}$$

  - Simpler and usually faster than RNC
  - MS Asure, Hadoop, OpenStack, etc

Distributed Storage System using Random Network Coding

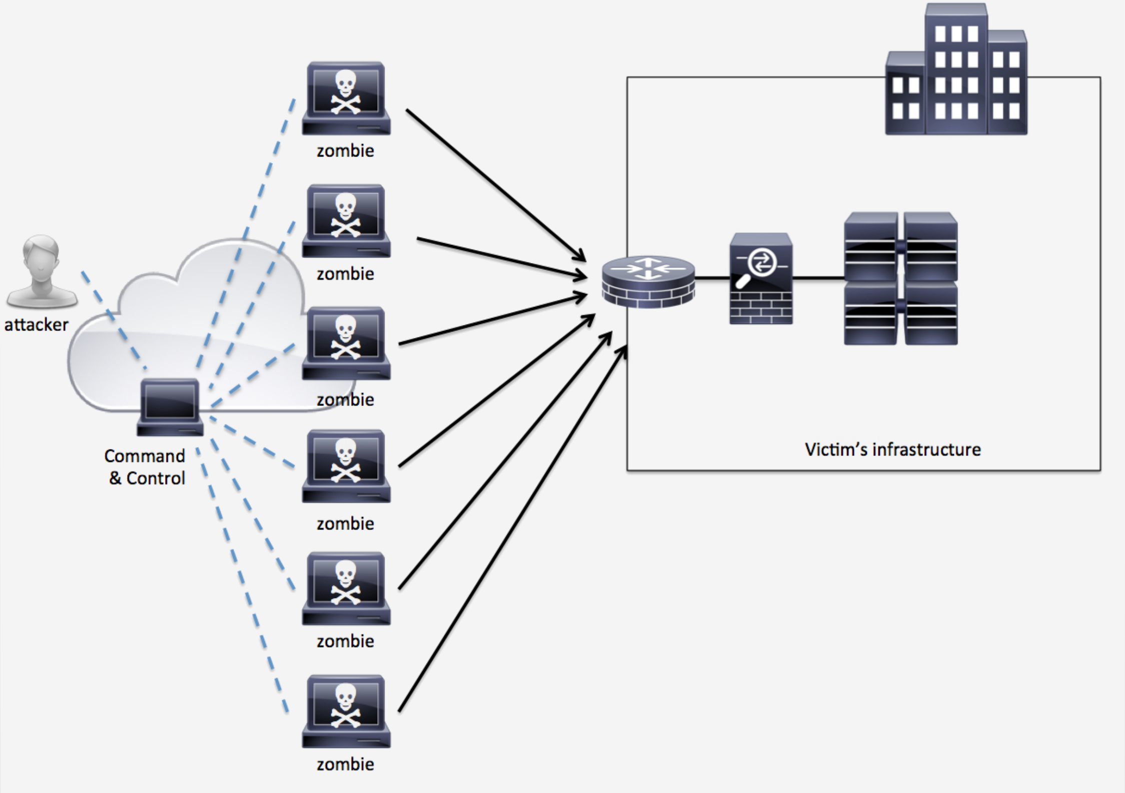
- Pros
  - Saves disk space
  - More reliable than traditional distributed system
  - Easy to add servers
  - Safe because data are encoded
- Cons
  - Encoding and decoding require CPU power
    - To solve linear equations, Gaussian Elimination is necessary ($O(n^3)$)
    - Calculation in GF is also slow?
    - Who decodes data?

Content Delivery Network (CDN)

- Puts the same contents on different servers
- Getting more popular

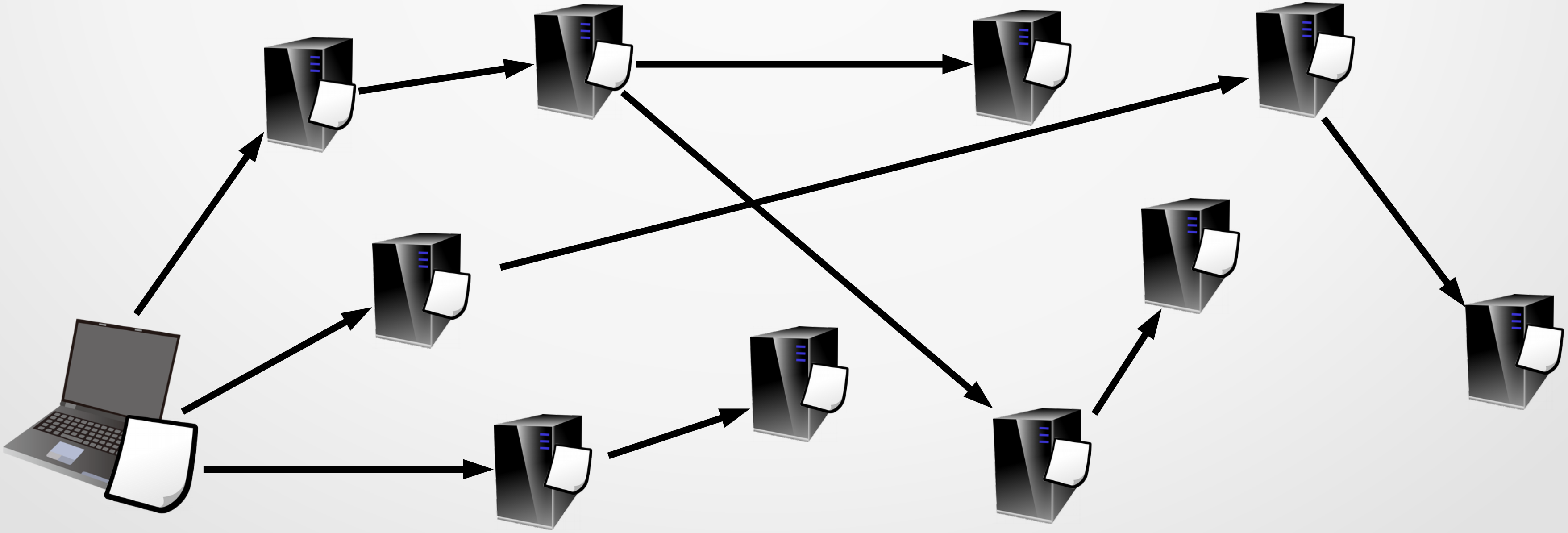
Content Delivery Network (CDN)

- DDoS attacks are increasing all over the world
- Enterprises employ temporary CDNs to survive attacks
- A DDoS attack costs only $5/h (free for first 5 min)



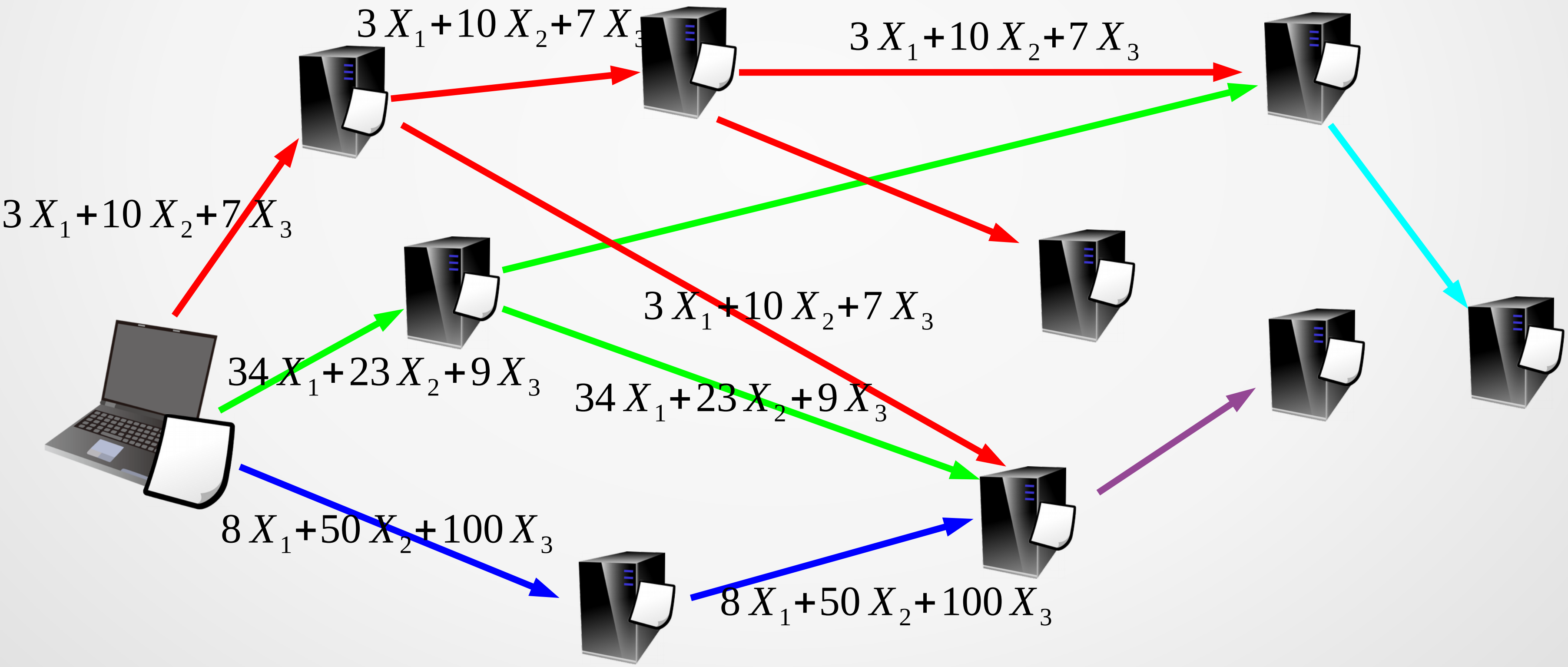
CDN + RNC

- Saves disk space – can utilize SSD space – achieves higher bandwidth
- But how do we distribute data?
- Can we reduce overall amount of transferred data?
- Can we guarantee non-duplication of equations?



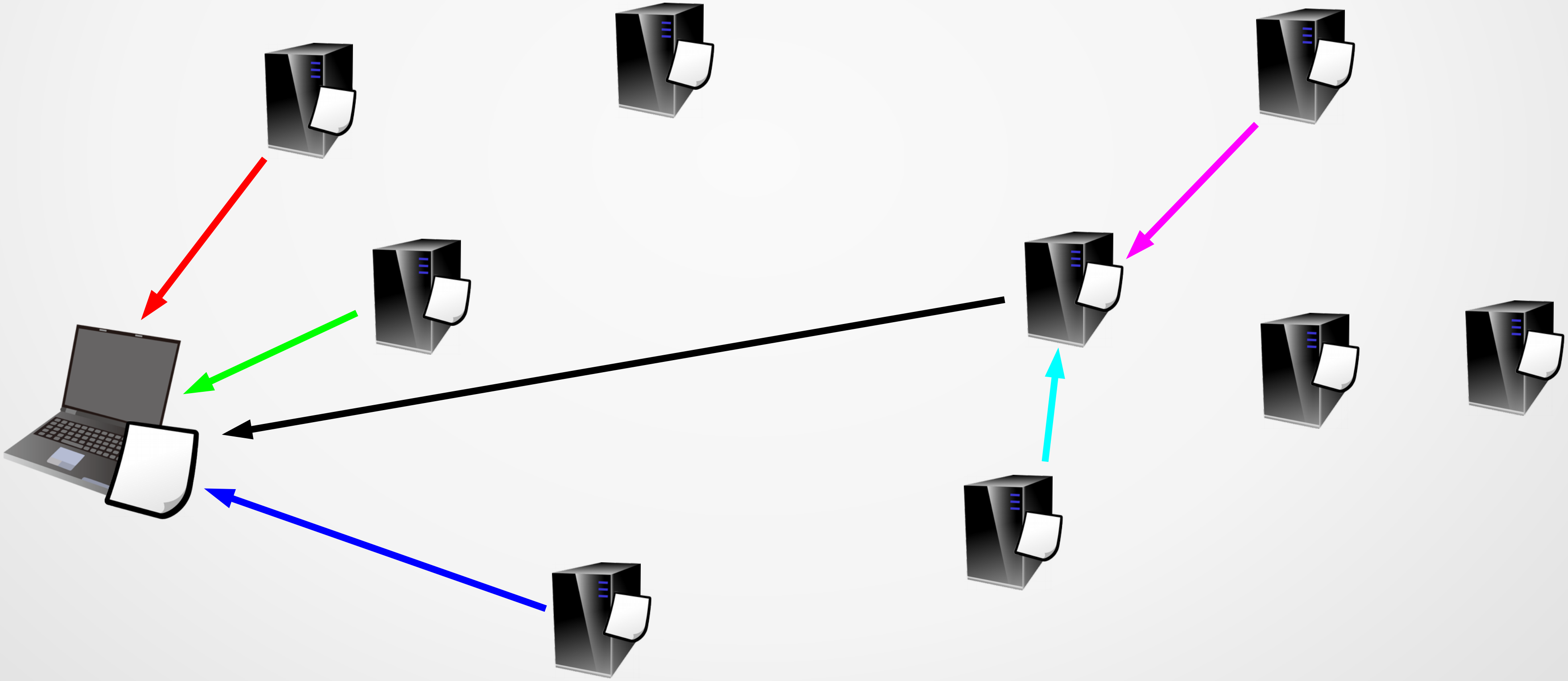
CDN + RNC

- How do we minimize duplication of equations?



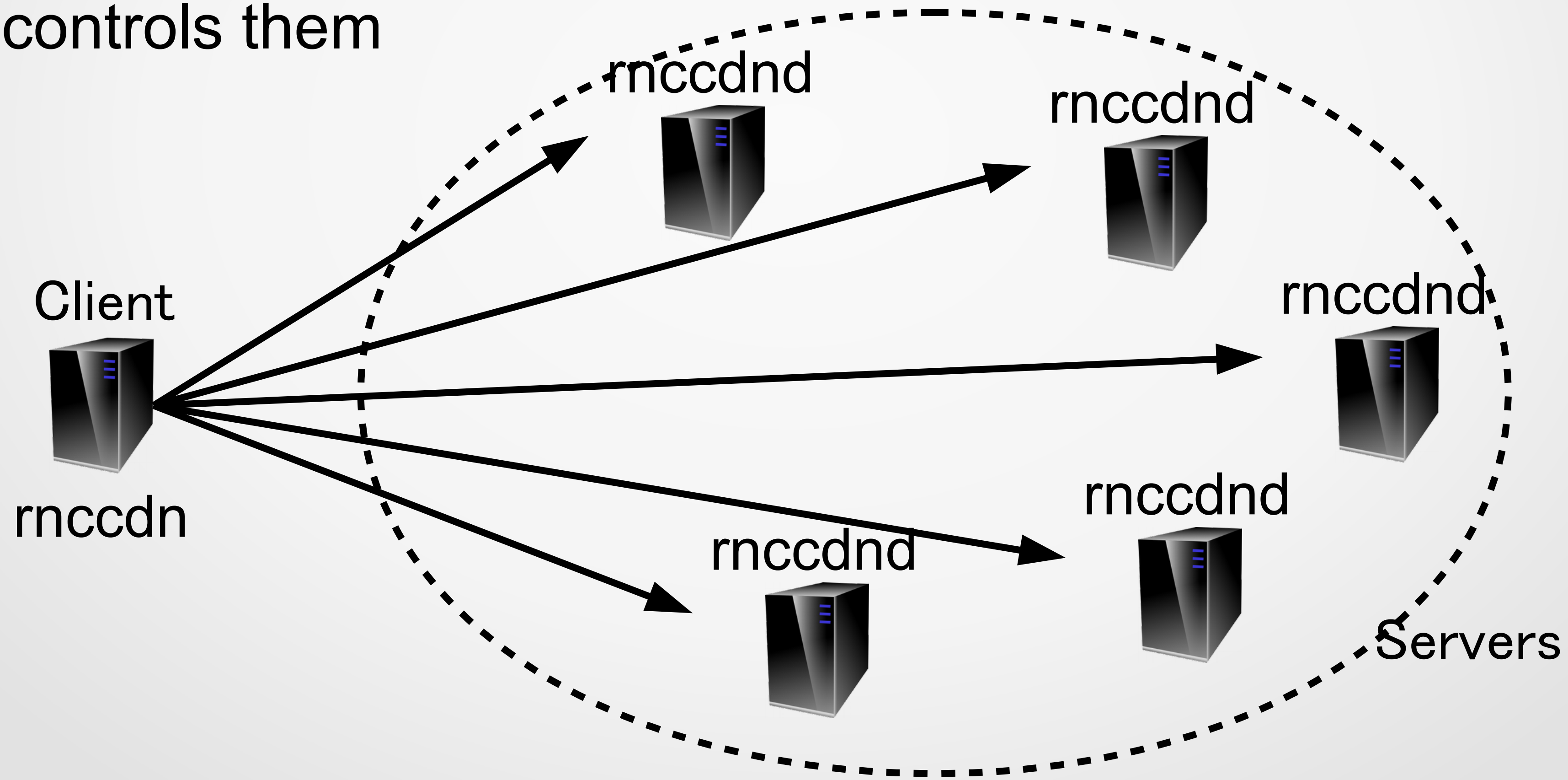
CDN + RNC

- Who decodes data? Client or server?
- Should we create plugin for web browsers?



Programs – rnccdn & rnccdnd

- Server: **rnccdnd** – daemon process – receives message from clients and other servers
- Client: **rnccdn** – sends requests + data to servers and controls them



Programs – rnccdn & rnccdnd

- Open source, BSD license (freer than GPL)
- Target OSs: Linux, FreeBSD
- Language: C or C++
- Libraries to use: libevent (optimizes polling functions), LibreSSL (for communication)?
- Message channel: SSL/TLS
- Data channel: SSL/TLS for raw data, non-encryption for encoded data
- HTTP/HTTPS for client–server communication?

Project Goal

- Creating open source programs that implement CDN + RNC
- If possible, implement a new technique to distribute encoded data