Estimation on Server Population with MLE-based CMR Model

WUNG, WEI-SHIANG 翁瑋襄
Network and Systems Laboratory
Graduate Institute of Electrical Engineering
National Taiwan University

Outline

- Introduction
- Server Population Estimation with Marked
- Data Set
- Experiment Design
- Experiment Result
- Conclusion

Introduction

- Twitch
- Motivation
- Problem Statement
- Contribution

Twitch

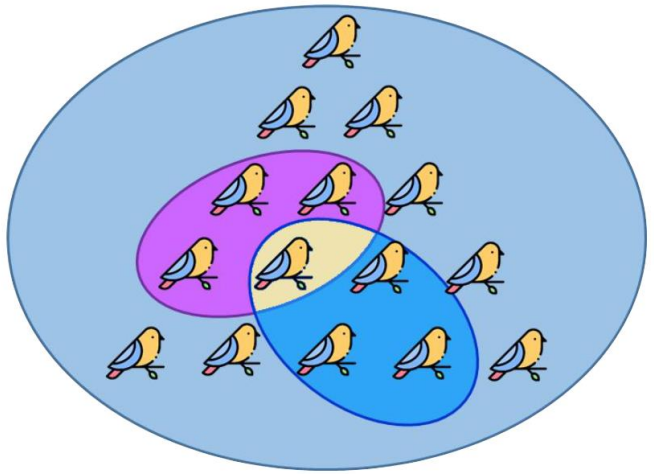
- Rapid growth of traffic volume since 2020
  - Due to the COVID-19 pandemic
- 43.6% of live video traffic
- Twitch's CDN
  - Most successful
  - Yet, little's known for its CDN size

Motivation

- Prior work to discover server population in Twitch's CDN
  - Repeatedly requesting video channels produces high probing overhead
  - One-time effort only
- Goal: continuously observe the CDN infrastructure
  - Sampling and estimating the CDN size with lightweight probing traffic!

Capture-Mark-Recapture (CMR)

- Similarity between animal population and server population in a CDN
  - Berth / death / move in / move out
- Estimate population with little effort
  - Examine the probability the marked animals being captured again
  - Mark animals in each capture



Our Previous Study in AINTEC

- Applying CMR to estimate server population in a CDN
  - Lincoln-Petersen (LP) Model
    - Closed population
  - Cormack-Jolly-Seber (CJS) Model
    - Allow varying survival rate and capturing probability over time
    - Open population

Cormack-Jolly-Seber (CJS) Model

- $\frac{R_t}{CN_t} = \frac{Z_t}{M_t - CM_t}$

  In the capture t Outside the capture t

- Probability of marked animals being caught in the future are identical

Relaxing Model Assumptions

- Restrictions of traditional CJS
  - All individuals share **the same capturing probability and survival rate**
  - Estimating the population size with history and future data
    - Offline estimation only
- Introduce **Maximum-Likelihood-Estimation-based CJS model with heterogeneity**

Maximum Likelihood Estimation (MLE)

- Determine values for the parameters of a model
  - **Maximize the likelihood** of the process described by **the model fitting the data**
- Log form of the probability model
  - Iteratively differentiate and update parameters

MLE-based CJS Prob. Model

$$\Pr(\mathrm{CH}_i \mid f_i) = \sum_{d=l_i}^{K} \left\{ \left(\prod_{j=f_i}^{d-1} \phi_j \right) (1-\phi_d) \left(\prod_{j=f_i+1}^{d} p_j^{x_{ij}} (1-p_j)^{1-x_{ij}} \right) \right\}$$

p_i: the capturing probability at the ith sample

ϕ_i: the survival rate from the ith to (i+1)th sample

Example:

$\Pr(111 \mid release\ at\ period\ 1) = \phi_1 \cdot p_2 \cdot \phi_2 \cdot p_3$

$\Pr(110 \mid release\ at\ period\ 1) = \phi_1 \cdot p_2 \cdot \chi_2,$
where $\chi_2 = (1-\phi_2) + \phi_2 \cdot (1-p_3)$

$\Pr(101 \mid release\ at\ period\ 1) = \phi_1 \cdot (1-p_2) \cdot \phi_2 \cdot p_3$

$\Pr(100 \mid release\ at\ period\ 1) = \chi_1,$
where $\chi_1 = (1-\phi_1) + \phi_1 \cdot (1-p_2)(1-\phi_2 \cdot p_3)$

$1 \xrightarrow{\phi_1} 2 \xrightarrow{\phi_2} 3$

$p_2 \quad p_3$

MLE-based CJS Model with Heterogeneity

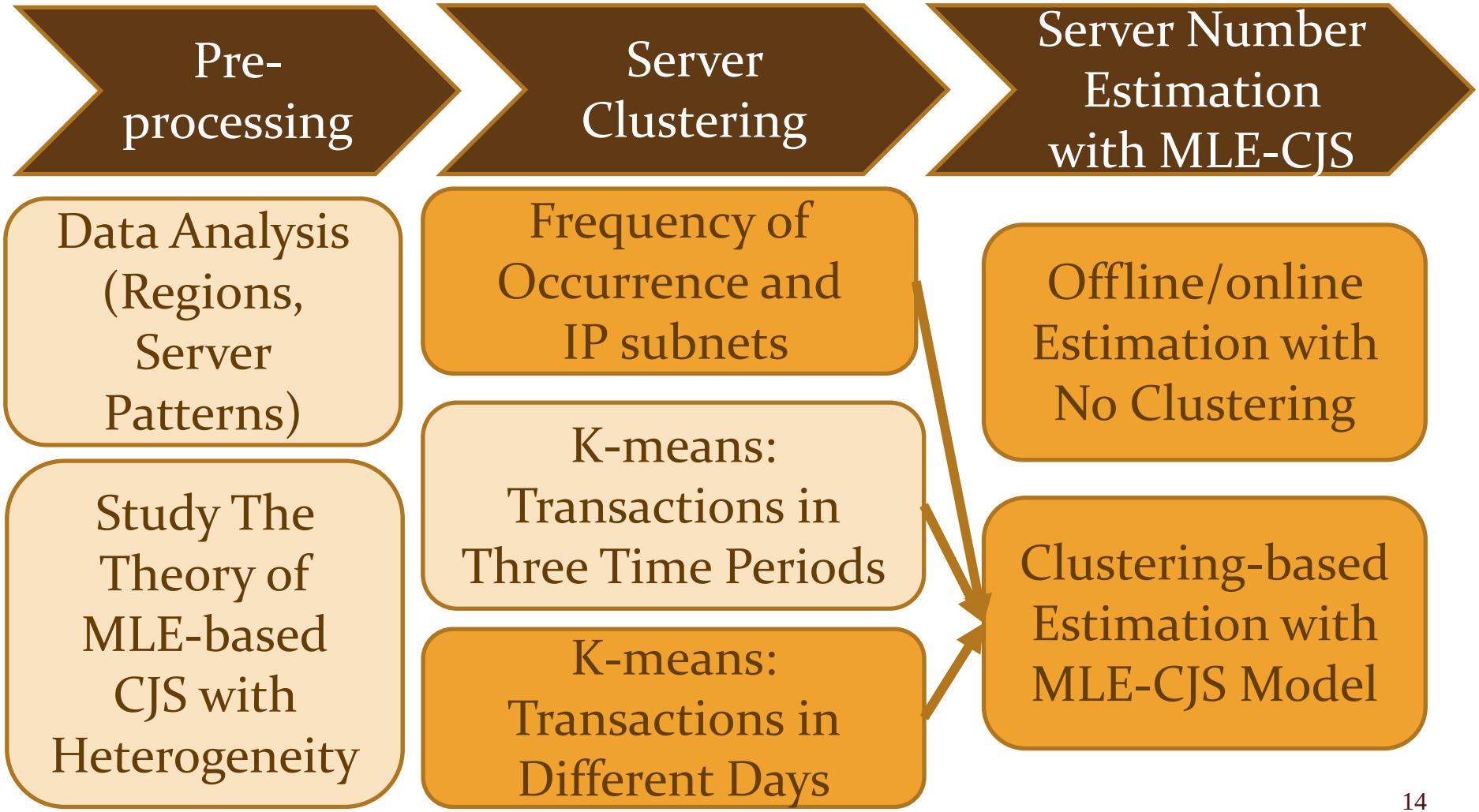
- Capturing probability and survival rate may be different for the animal groups.
- $logit(p_{jc}) = \log\left(\frac{p_{jc}}{1-p_{jc}}\right) = \mu + \tau_j + \eta_c + (\tau\eta)_{jc}$
- Similar to ϕ_{jc}
- Solved p_{jc} and ϕ_{jc} by MLE

- μ is the average
- τ_j & η_c are the variance terms for sample time and individual groups
- $(\tau\eta)_{jc}$ is the covariance term for sample time and individual groups

Problem Statement

- **Maximum-Likelihood-Estimation-based CJS model**
  - Co-estimating all parameters of the CJS probability model
    → higher level of accuracy with the same amount of data
  - Allowing individuals with different capturing probability and survival rate (**heterogeneity**)
    → improve estimation accuracy
- **Server clustering**
  - Reduce the dimension of CJS probability model
    → lower computational overhead

Co-work with Hsu Cheng



Contribution

- MLE-based model can achieve a higher level of accuracy with the same amount of data
- Estimate server population with heterogeneity
- Identify the detail reasons to explain why clustering-based estimations lead to worse results.

Server Population Estimation with Marked

- Introduction to marked
- Data preprocessing
- Capturing Probability and Survival Rate Estimation
- Server Population Estimation

Marked

- An R package
- Implementation of MLE-based CJS model with heterogeneity
- Co-estimate capturing probability and survival rate in each server cluster
- Ref: https://cran.r-project.org/web/packages/marked/index.html

Data Preprocessing

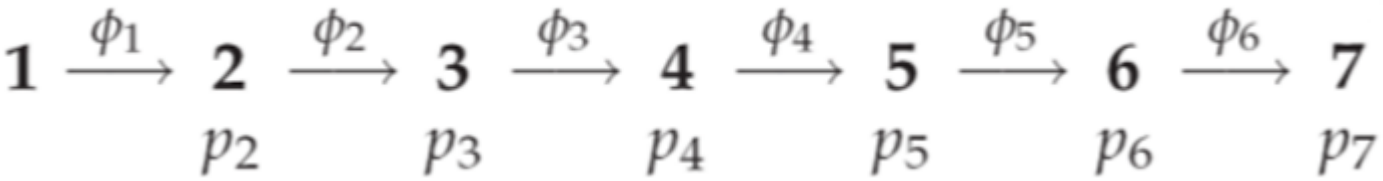
- Each row represents a server IP
- For each server IP
  - Capturing history
  - Group label

	ch	ip_freq
1	11100001100010000011001000001001001000101010	mid
2	00000001000000000000000000000000000000000000	low
3	00010101100000001011001000001001100000100 1	mid
4	00010101100001001010100000001001001000101 1	mid
5	11111111111111111111111111111111111011111	high
6	11111111111111111111111111111111111011111	high
7	00000001100000001010100100001001101000100 1	mid
8	11111111111111111111111111111111111011111	high
9	00110101100000001011000000001001101000101 1	mid
10	10000101100000000000010000001001001000101 1	mid

Model Fitting

- "process.data()"
  - Give input data
  - Define group labels
- "make.design.data()"
  - Create design data frames for model fitting
- "crm()"
  - Fit a CJS model with the design data frames
  - Optimize with Maximum-Likelihood-Estimation

Capturing Probability and Survival Rate



- ϕ_i is the survival rate from the ith sample to the $(i+1)$th sample.
- p_i is the capturing probability of the ith sample.

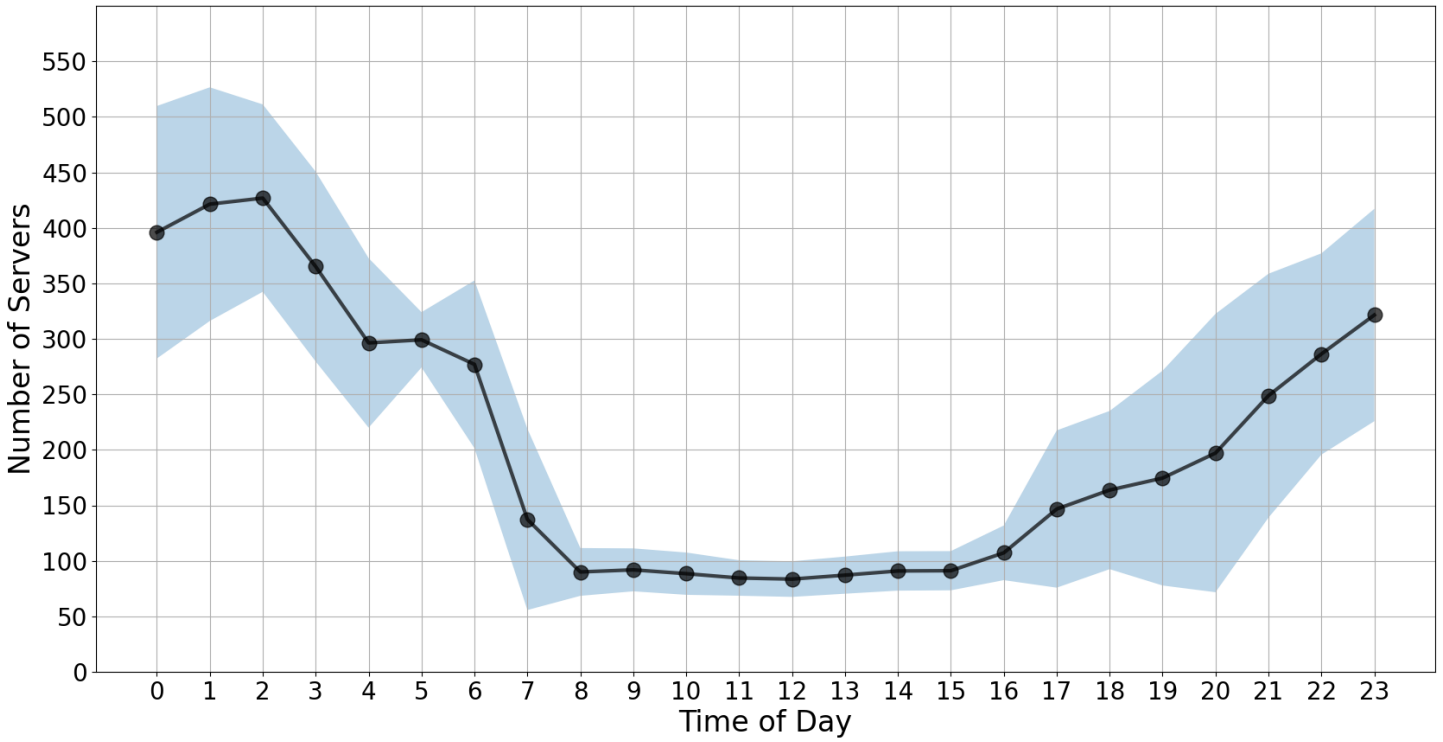
Server Population Estimation

- Server population * capturing probability = numbers of servers being captured
- Server population = numbers of servers being captured / capturing probability

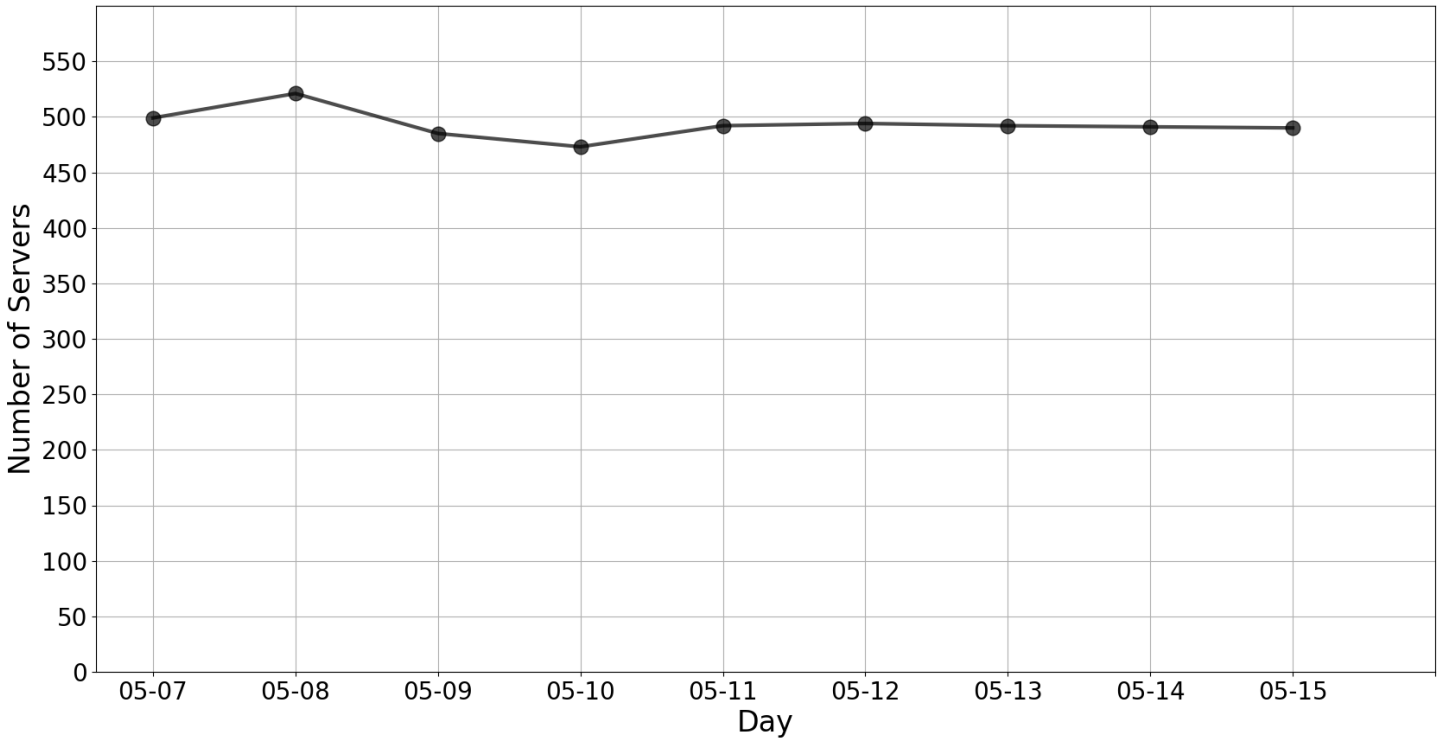
Data Set

An US Data Set Collected by Caleb Wang in May 2021

Hourly Server Count



Daily Server Count



Experiment Design

- Offline/online estimation
- Clustering-based estimation

Offline / Online Estimation

- **Not consider heterogeneity**
- Offline Estimation

- Online Estimation
  - Real time

  - One day delay

Server Clustering

- **Consider heterogeneity**
- Clustering strategies
  - Frequency of occurrence
  - Server IP prefix
  - K-means clustering with transaction numbers
- Overall server population = sum of server population in each group

Frequency of Occurrence

- Core servers and supportive servers
- Numbers of appearing time / numbers of samples
  - ≥ 1/2 → high frequency
  - < 1/5 → low frequency
  - Else → medium frequency

Server IP Prefix

- Servers with same IP prefixes → similar behavior
- Short IP prefixes → coarse server clustering
- Long IP prefixes → fine server clustering
  - Small server numbers in each group → estimation divergence
- We try 16-bit and 24-bit prefixes for clustering

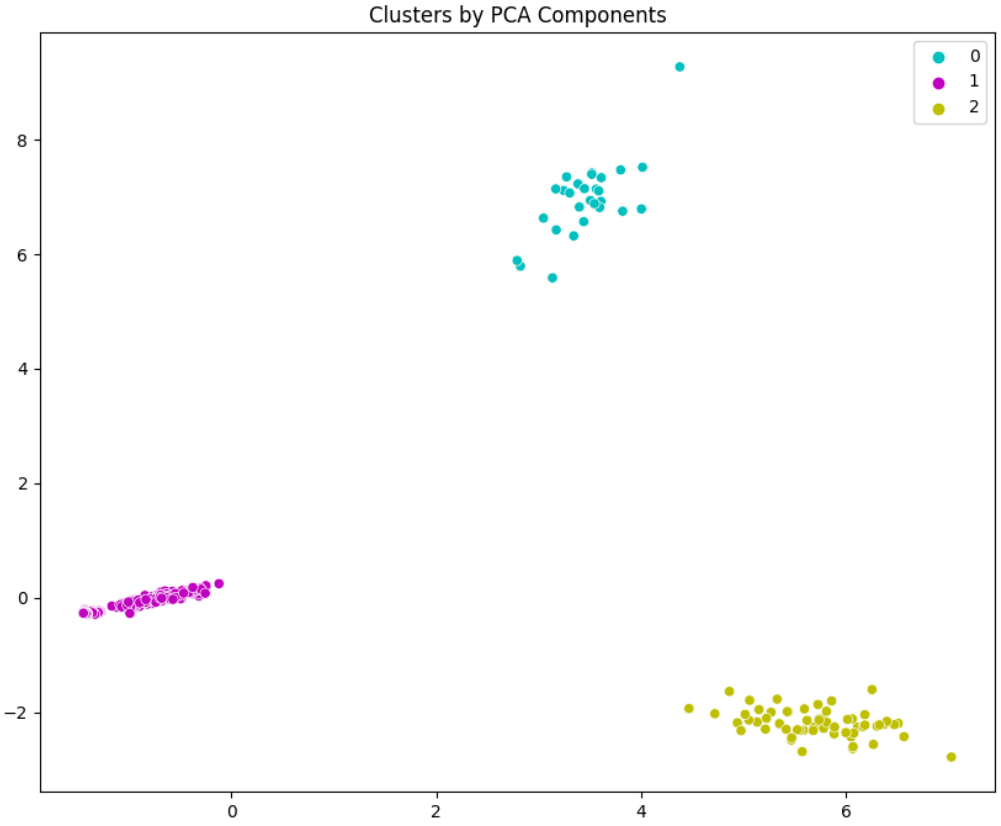
K-means Clustering

- Phase 1: (borrowing from Hsu Cheng's work)
  - 3 server groups (orange, blue, green)
  - Servers in the blue group do not have regular patterns
- Phase 2:
  - Further evaluations on the blue group
  - Clustering by the transaction numbers in each days
  - 5 groups in total (2 in phase 1 + 3 in phase 2)

Clustering By Transaction Numbers in Different Days (Phase 2)

- The transaction numbers in each day (5/7~5/15, 9 days)
- Transform into a 9-dimension vector for clustering
- Example:
  - '52.223.227.169': [9283, 9765, 7412, 533, 731, 556, 706, 737]
  - '99.181.97.72': [125, 85, 4, 97, 76, 16, 57, 125]

K-means Clustering by Transaction Numbers (Phase 2)



Characteristics of the Three Server Groups (Phase 2)

- Cyan group (28 servers, sharing IP prefix '52.223.227')
  - More transactions in the first few days
  - [9283, 9765, 7412, 1959, 533, 731, 556, 706, 737]
- Magenta group (408 servers)
  - Occasionally showing up
  - [125, 345, 85, 4, 97, 76, 16, 57, 125]
- Yellow group (61 servers, sharing IP prefix '192.16.65')
  - More transactions in the last few days
  - [37, 113, 25, 1677, 2484, 4408, 4540, 3230, 3063]

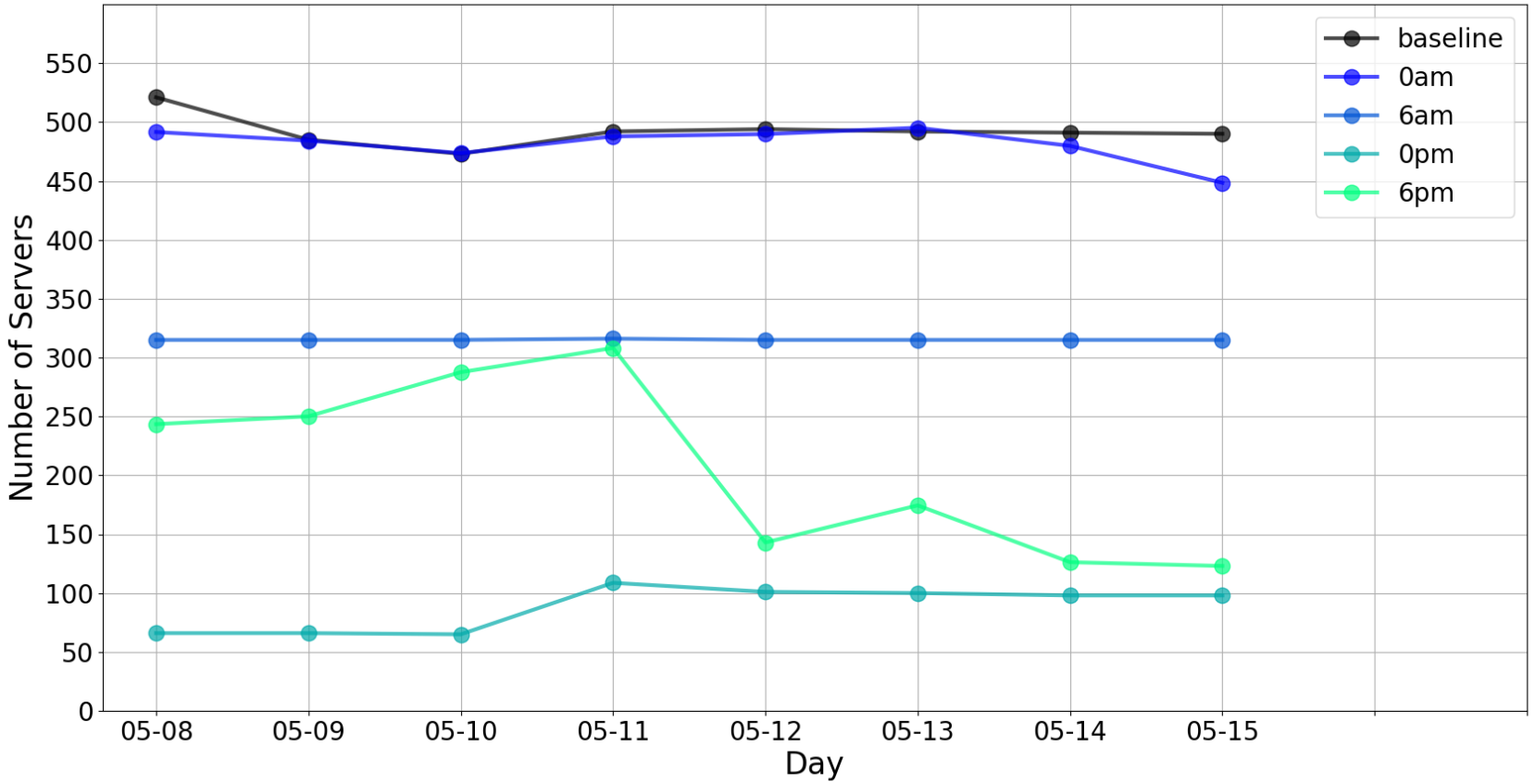
Experiment Results

- Offline/online estimation
- Clustering-based estimation

Offline Estimation – Model Comparison (No heterogeneity)

Sample Time	Traditional CJS Model	MLE-based CJS model
0-1 am	1.19%	0.82%
6-7 am	35.38%	35.38%
0-1 pm	81.70%	81.64%
6-7 pm	55.59%	55.81%

Offline Estimation – MLE-based



Online Estimation – Real Time

(No heterogeneity)

- Average error rates
  - Traditional CJS: 15.02% / MLE-based CJS: 22.61%
- Reasons
  - Traditional CJS estimates population with history and future records
  - MLE-based CJS cannot estimate survival rate in real time → inaccurate capturing probability

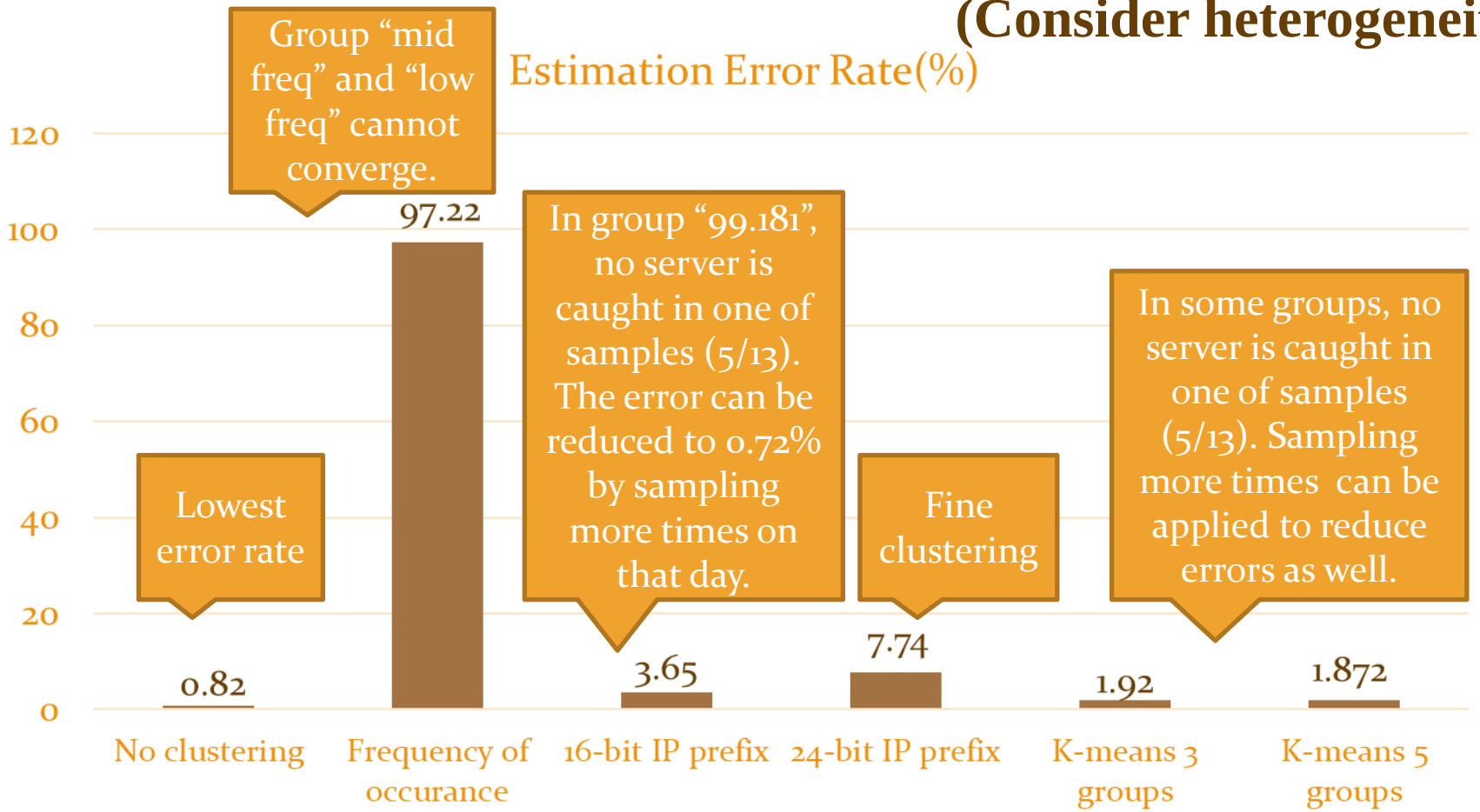
Online Estimation – 1 Day Delay

(No heterogeneity)

- Average error rates
  - Traditional CJS: 5.04% / MLE-based CJS: 3.64%
- Reasons
  - Traditional CJS significantly relies on future records → The estimation result does not converge
  - MLE-based CJS is able to co-estimate capturing probability and survival rate at the sample point

Clustering-based Estimation

(Consider heterogeneity)



Conclusions

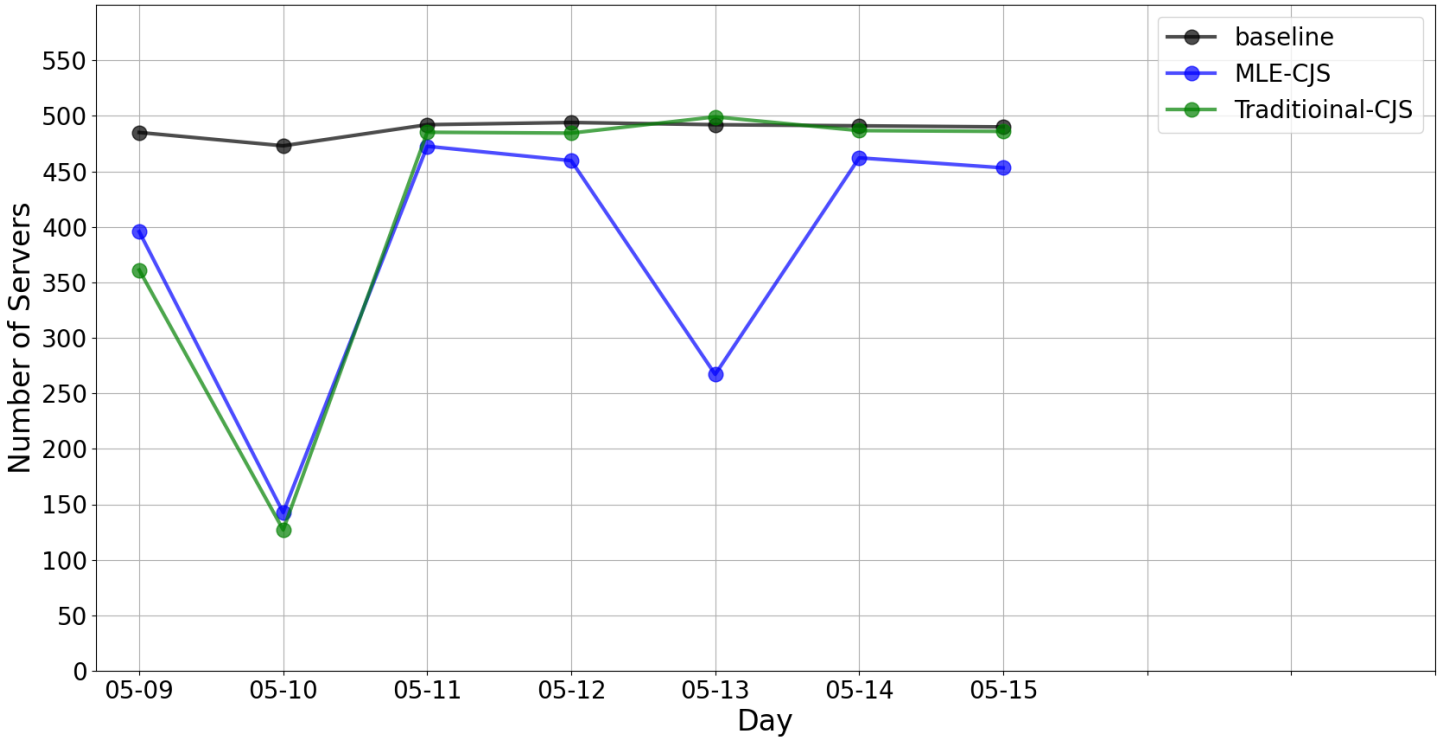
- This study aims to improve server population estimation accuracy and achieve a higher accuracy in online estimation by MLE-based CJS model with heterogeneity. Server clustering strategies are applied to reduce the computational overhead.
- MLE-based model indeed has a better online estimation result with one day delay.
- Although the estimation of server clustering does not perform better than no clustering, we discover the detail reasons by analyzing the estimation results in each server group.

Q&A

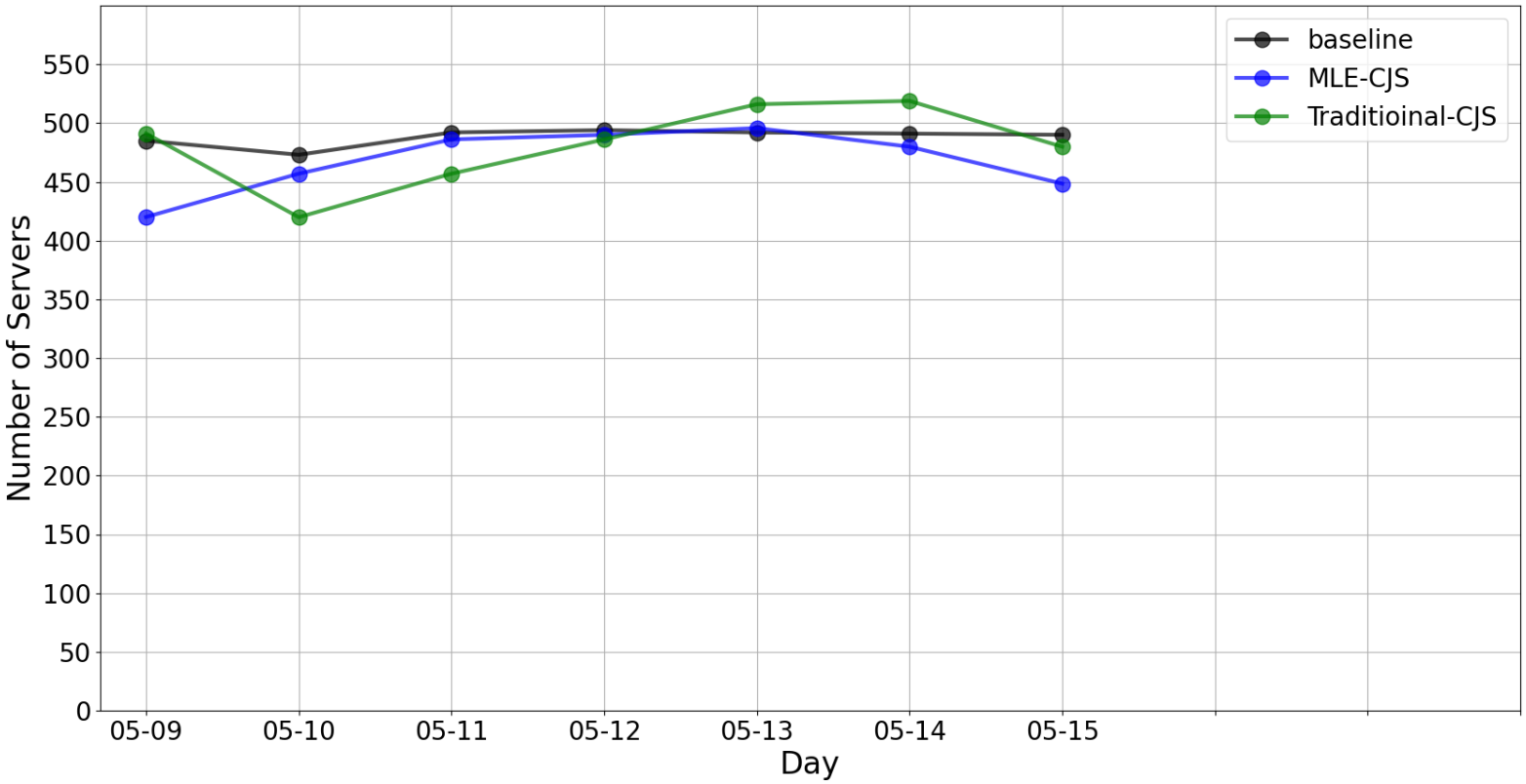
Appendix

- Daily estimation results (real time / delay one day)
- Convergence Periods of the Two CJS Models
- Clustering by frequency of occurrence
- Clustering by IP prefix
- K-means clustering by transaction numbers
- Improve accuracy with extra samples

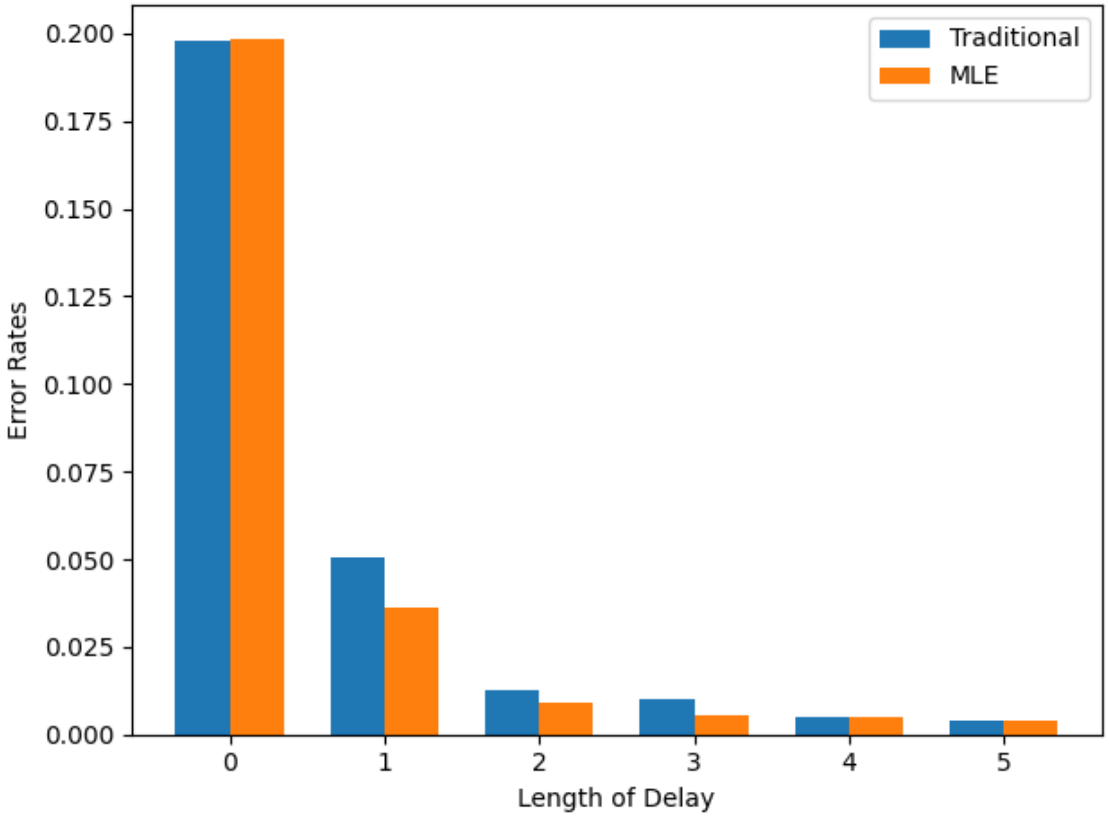
Daily Estimations (Real Time)



Daily Estimations (Delay One Day)



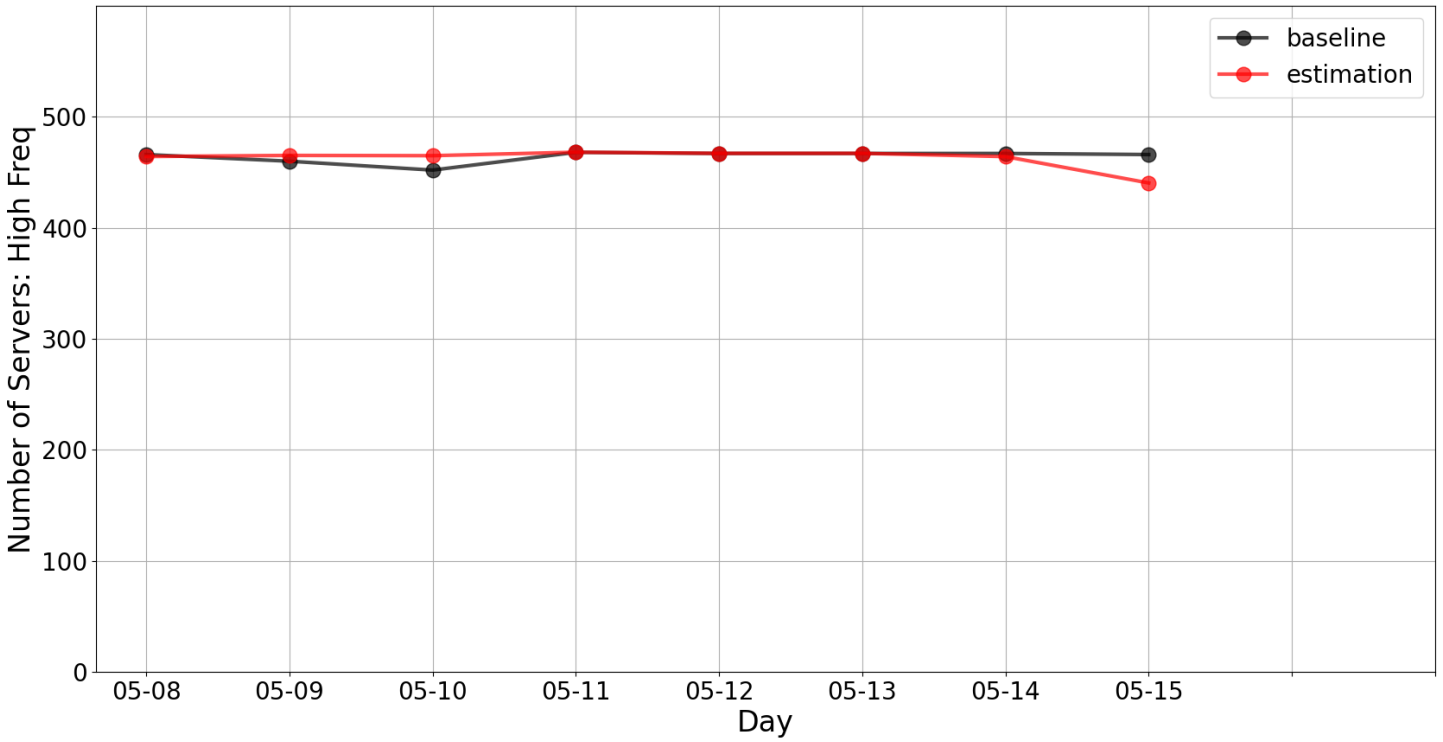
Convergence Periods of the Two CJS Models



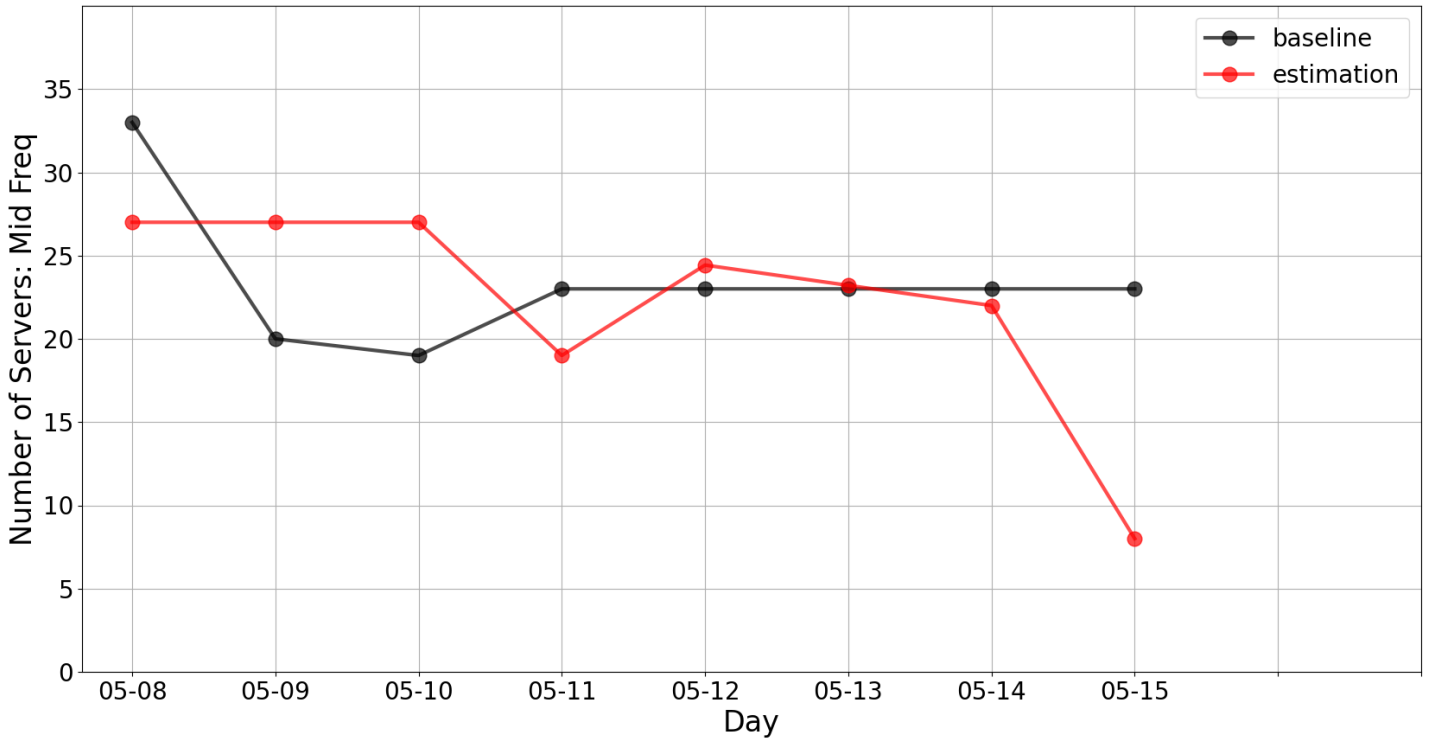
Clustering by Frequency of Occurrence

Frequency	Server numbers	Average estimation error (%)
Total	521	97.22
High	468	0.78
Mid	45	17.66
Low	8	20494.97

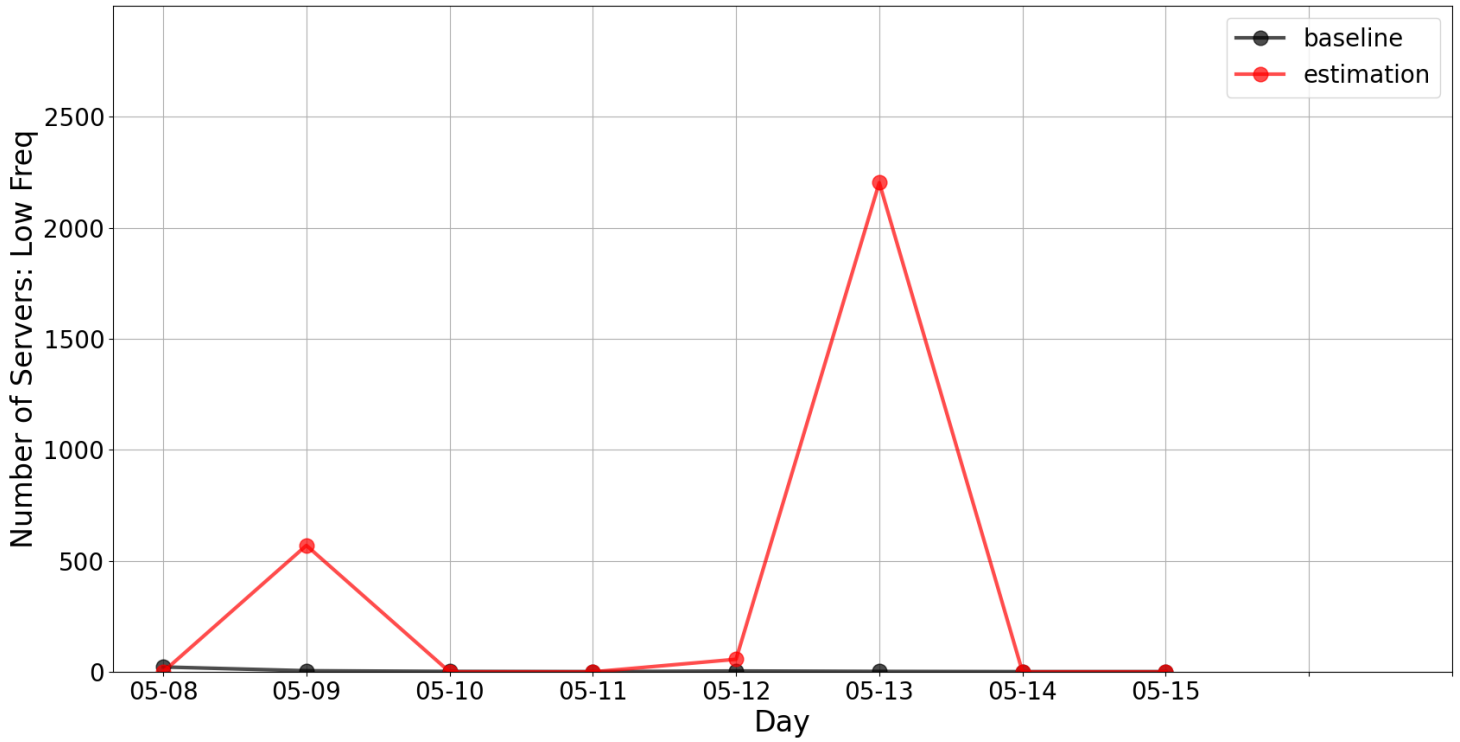
Estimation Result (“High Freq”)



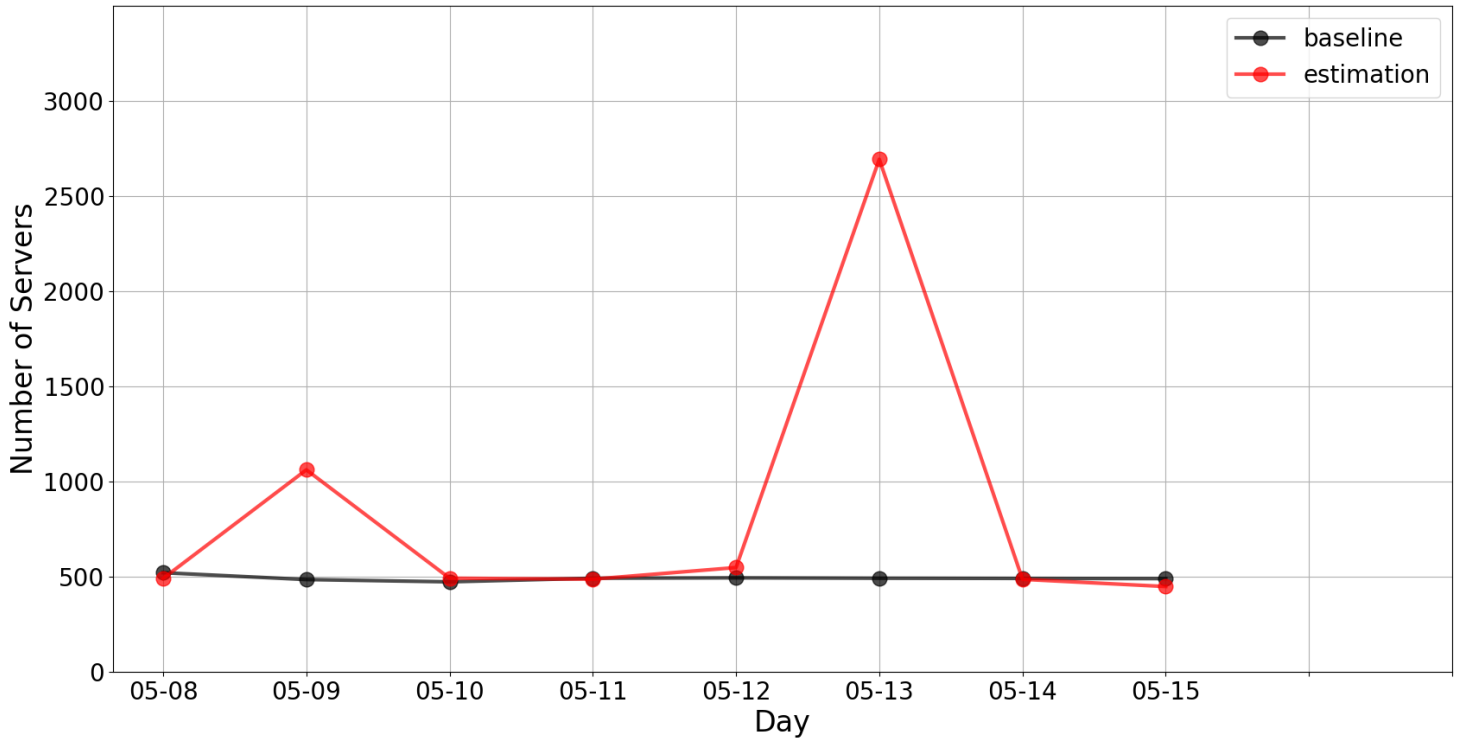
Estimation Result (“Mid Freq”)



Estimation Result ("Low Freq")



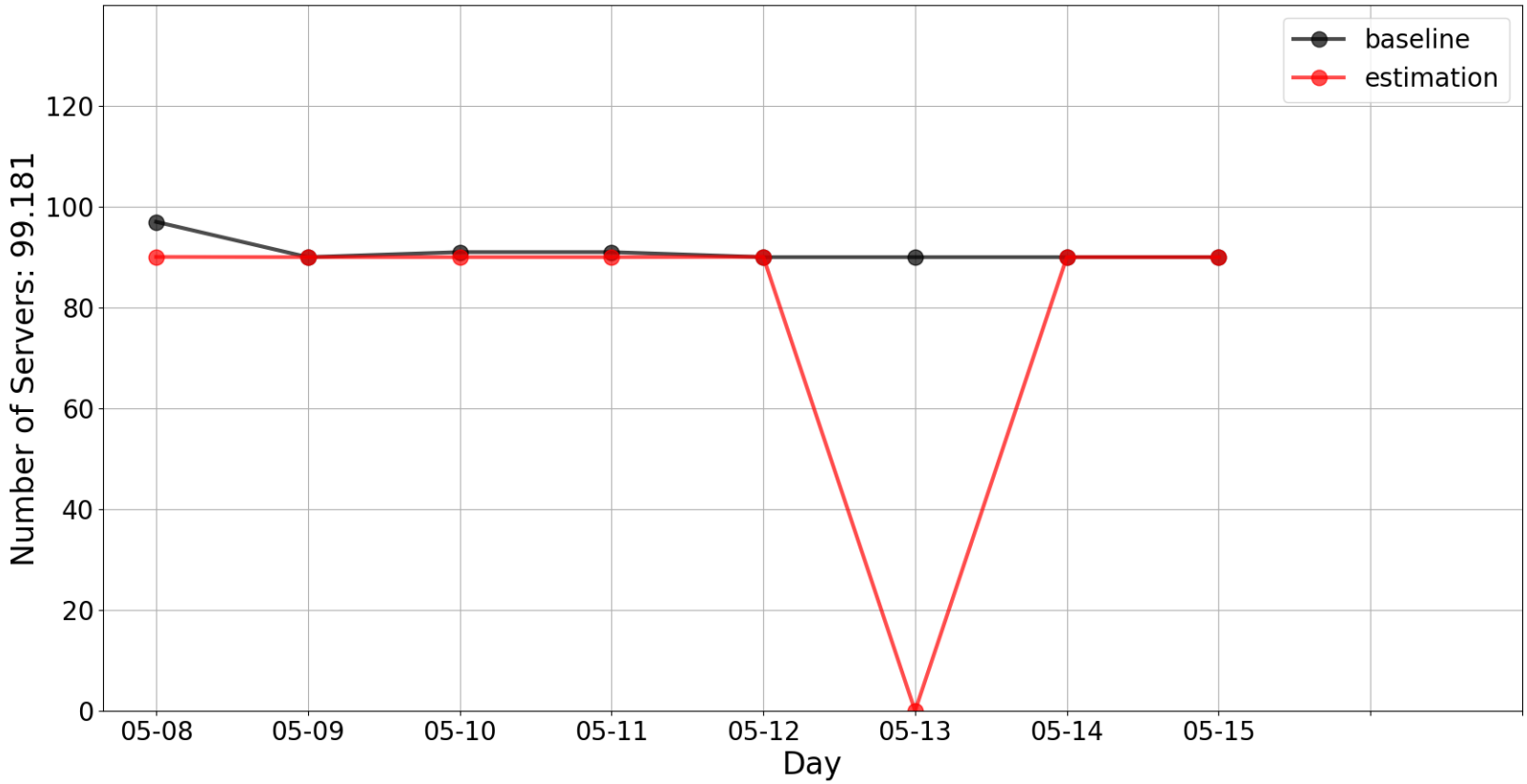
Estimation Result (Total)



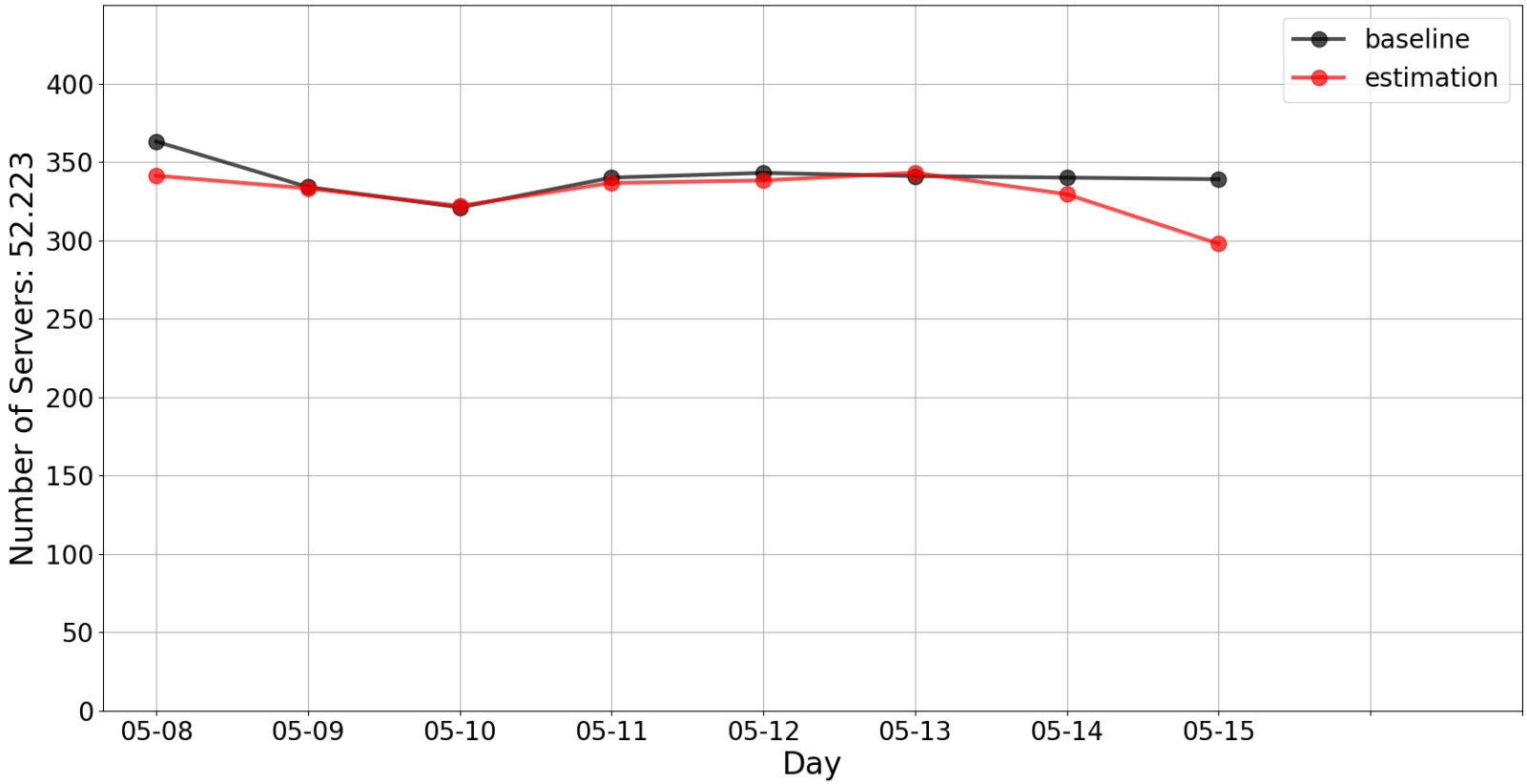
Clustering by 16-bit IP Prefix

16-bit IP prefix	Server numbers	Average estimation error (%)
Total	538	3.65
99.181	98	17.04
52.223	379	1.11
192.16	61	0.15

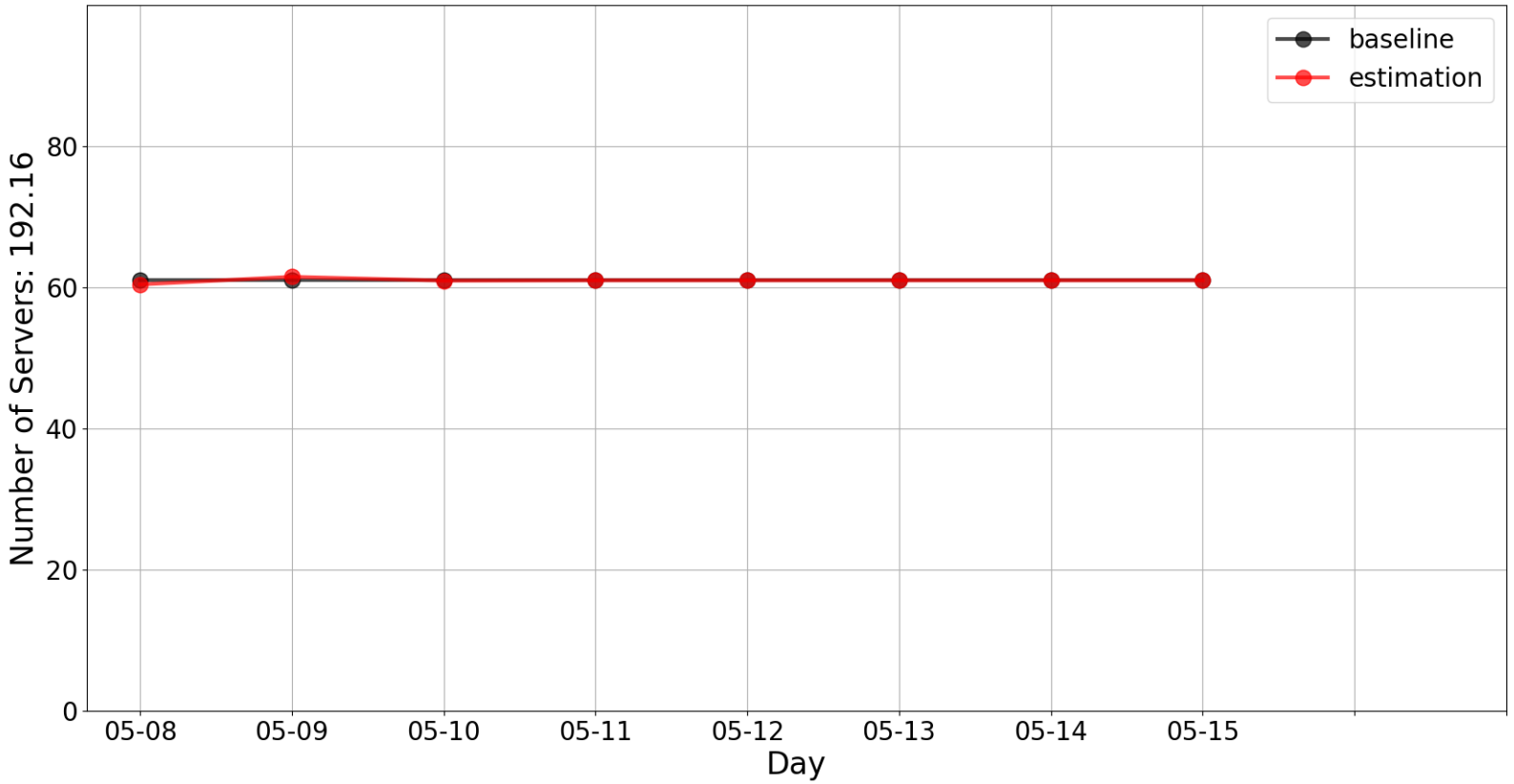
Estimation Result (“99.181”)



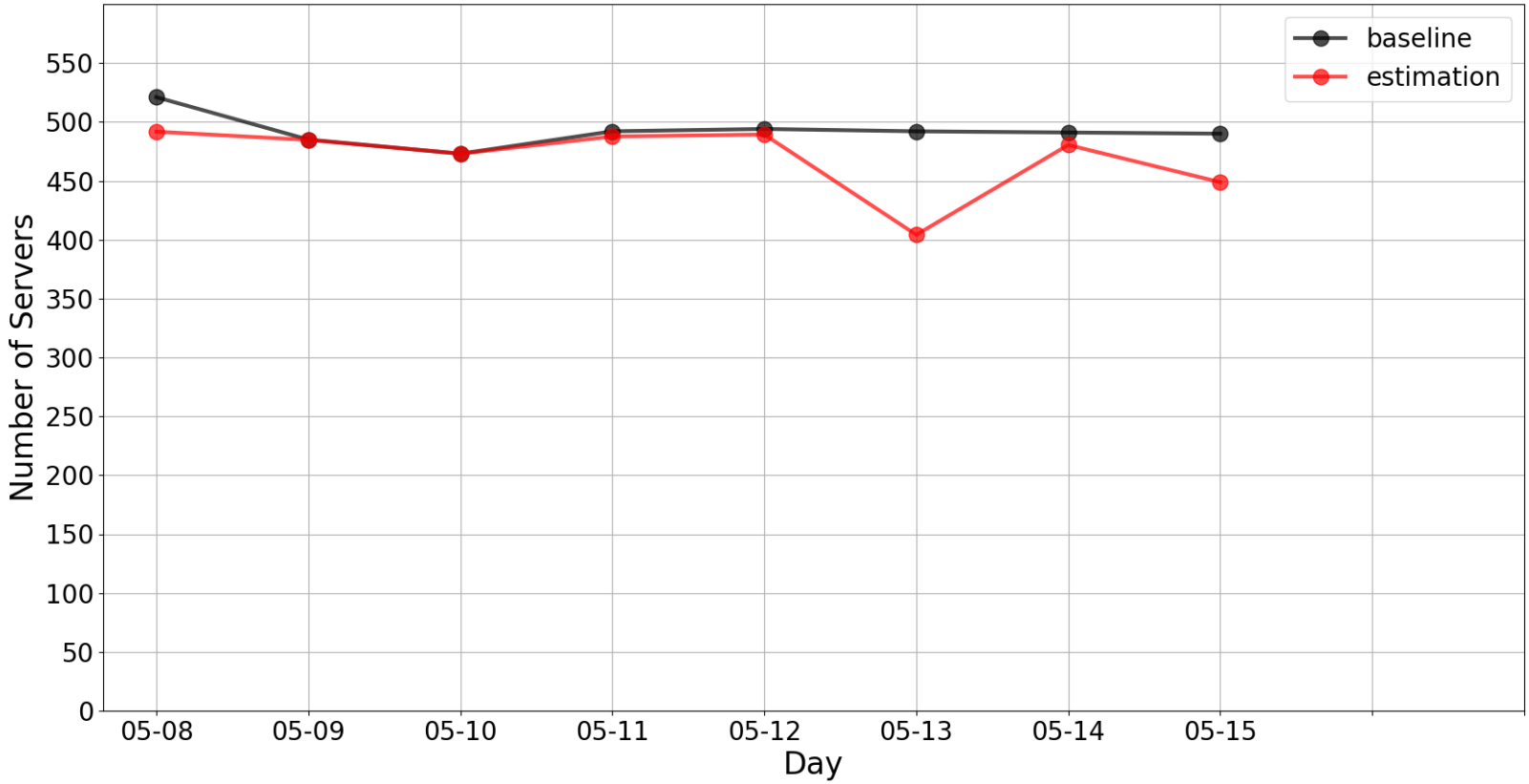
Estimation Result (“52.223”)



Estimation Result (“192.16”)



Estimation Result (Total)

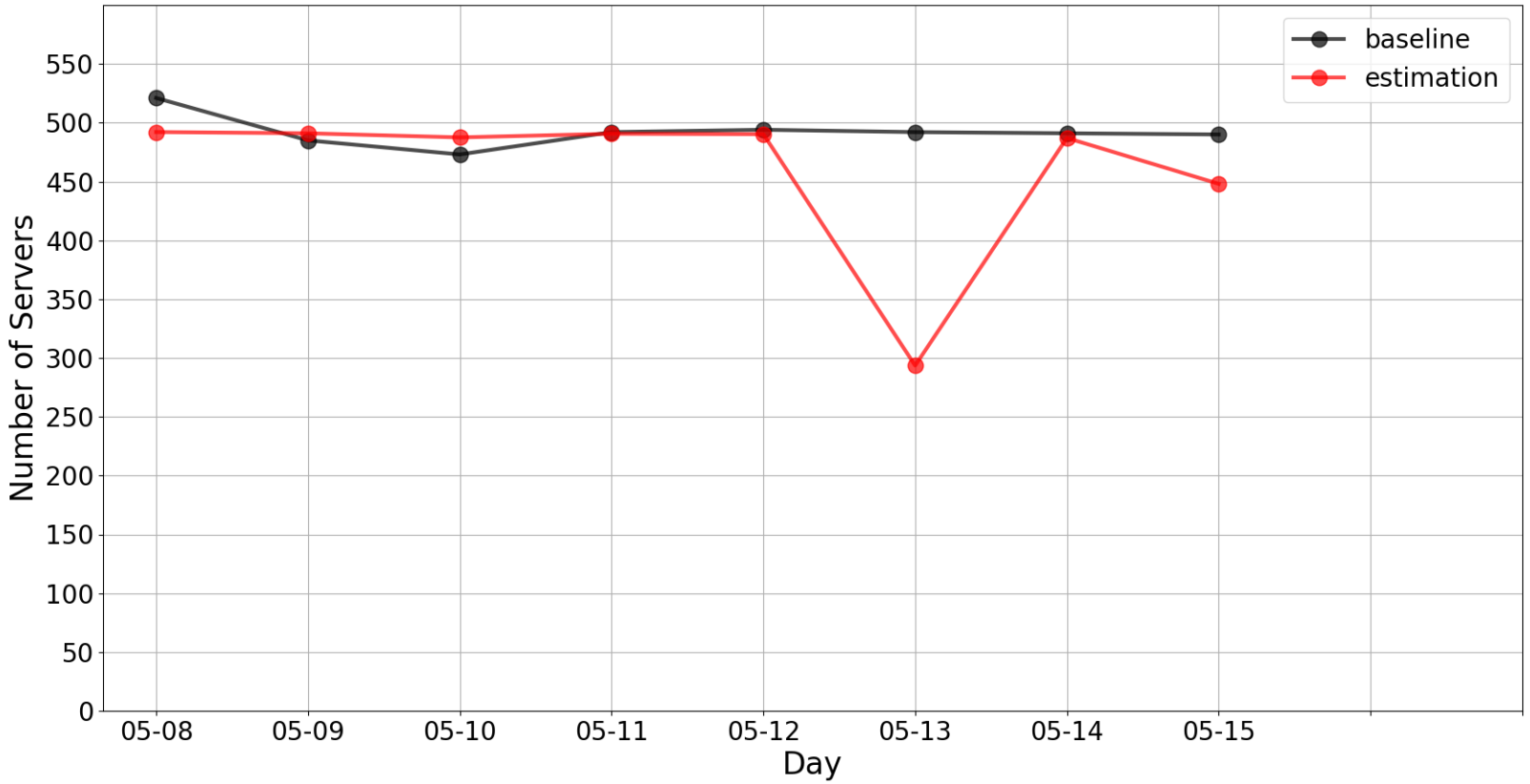


Clustering by 24-bit IP Prefix

24-bit IP prefix	Server numbers	Average estimation error (%)
Total	538	7.74
99.181.96	16	16.67
52.223.226	68	16.67
99.181.97	81	17.09
52.223.244	41	16.67
52.223.227	31	0
52.223.224	19	0
52.223.229	12	0

24-bit IP prefix	Server numbers	Average estimation error (%)
52.223.246	35	0.49
52.223.228	39	0
192.16.65	61	0.14
52.223.243	97	7.32
52.223.225	34	0.56
99.181.65	1	12.50
52.223.247	2	12.50
52.223.248	1	12.50

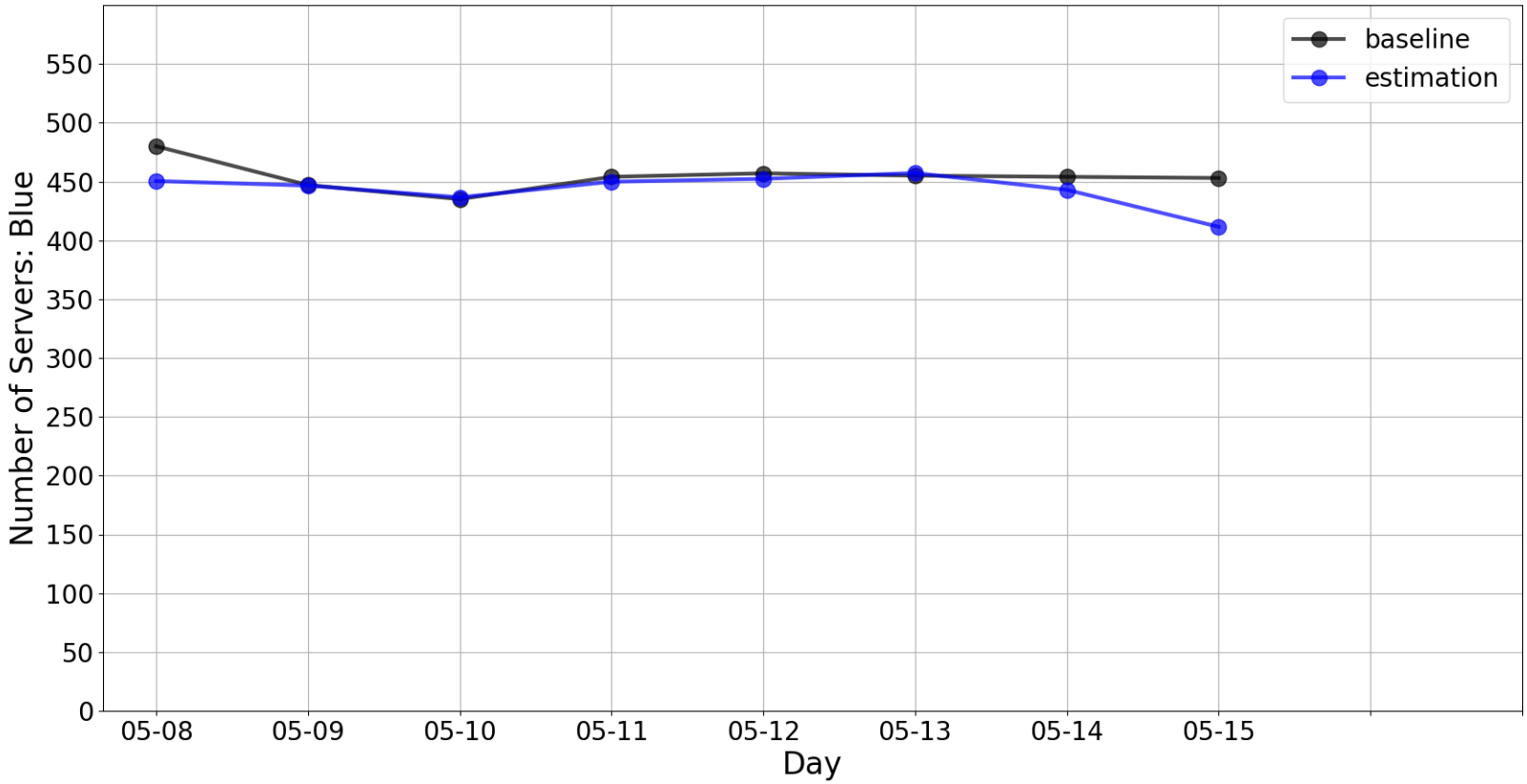
Estimation Result (Total)



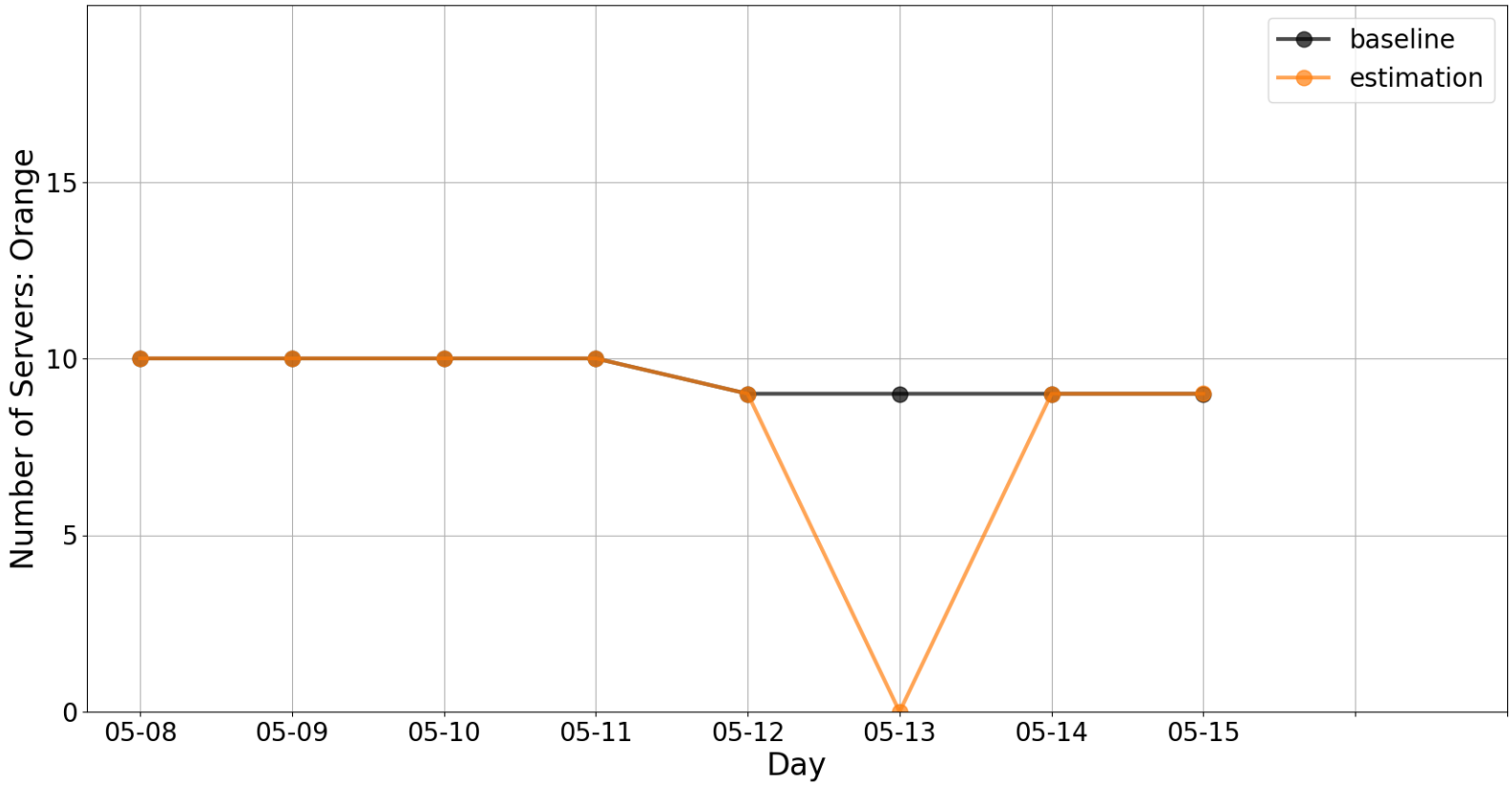
K-means Clustering Phase 1

Group	Server numbers	Average estimation error (%)
Total	538	1.92
Blue	497	0.89
Orange	10	16.67
Green	31	16.67

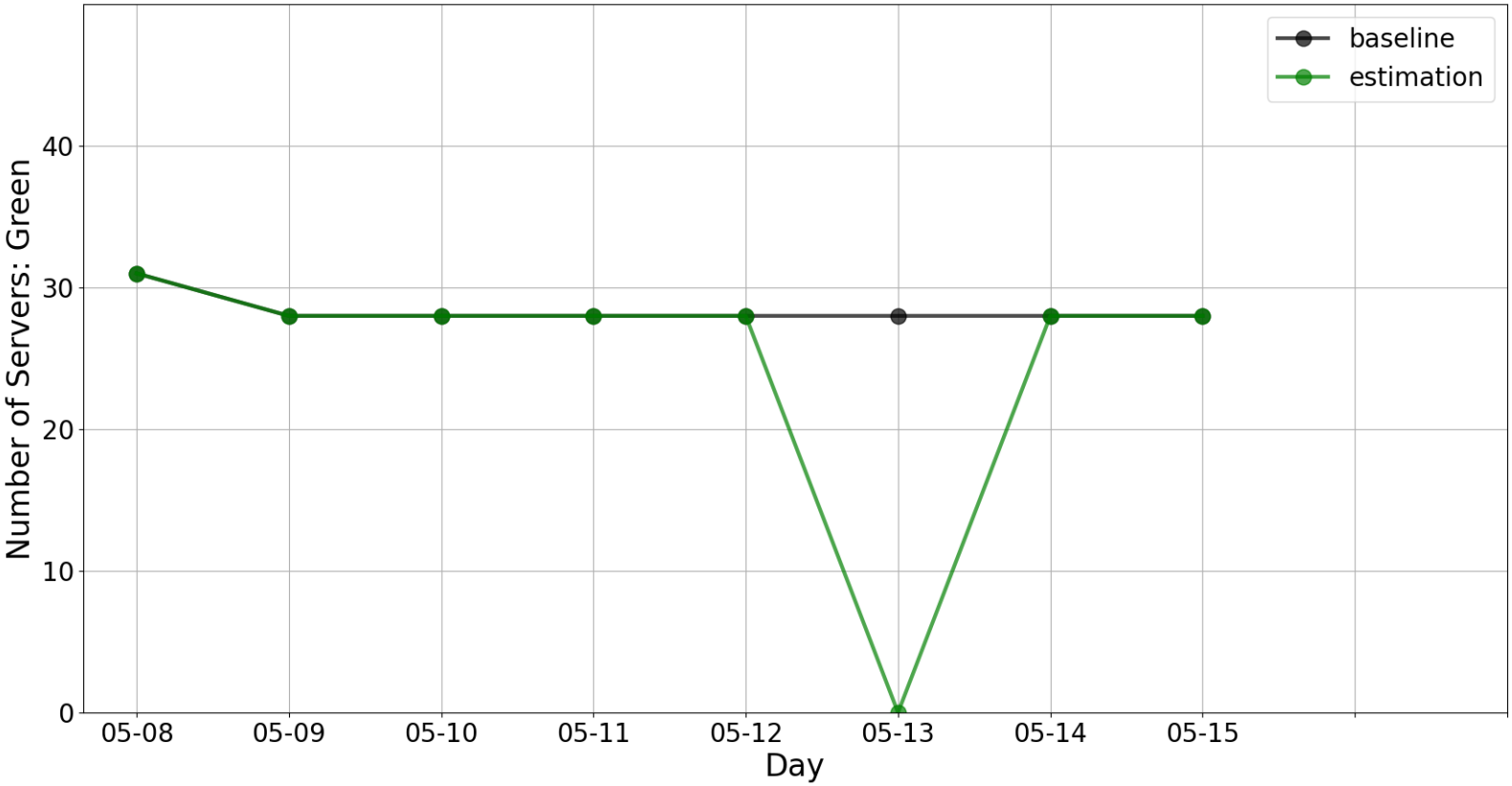
Estimation Result (Blue)



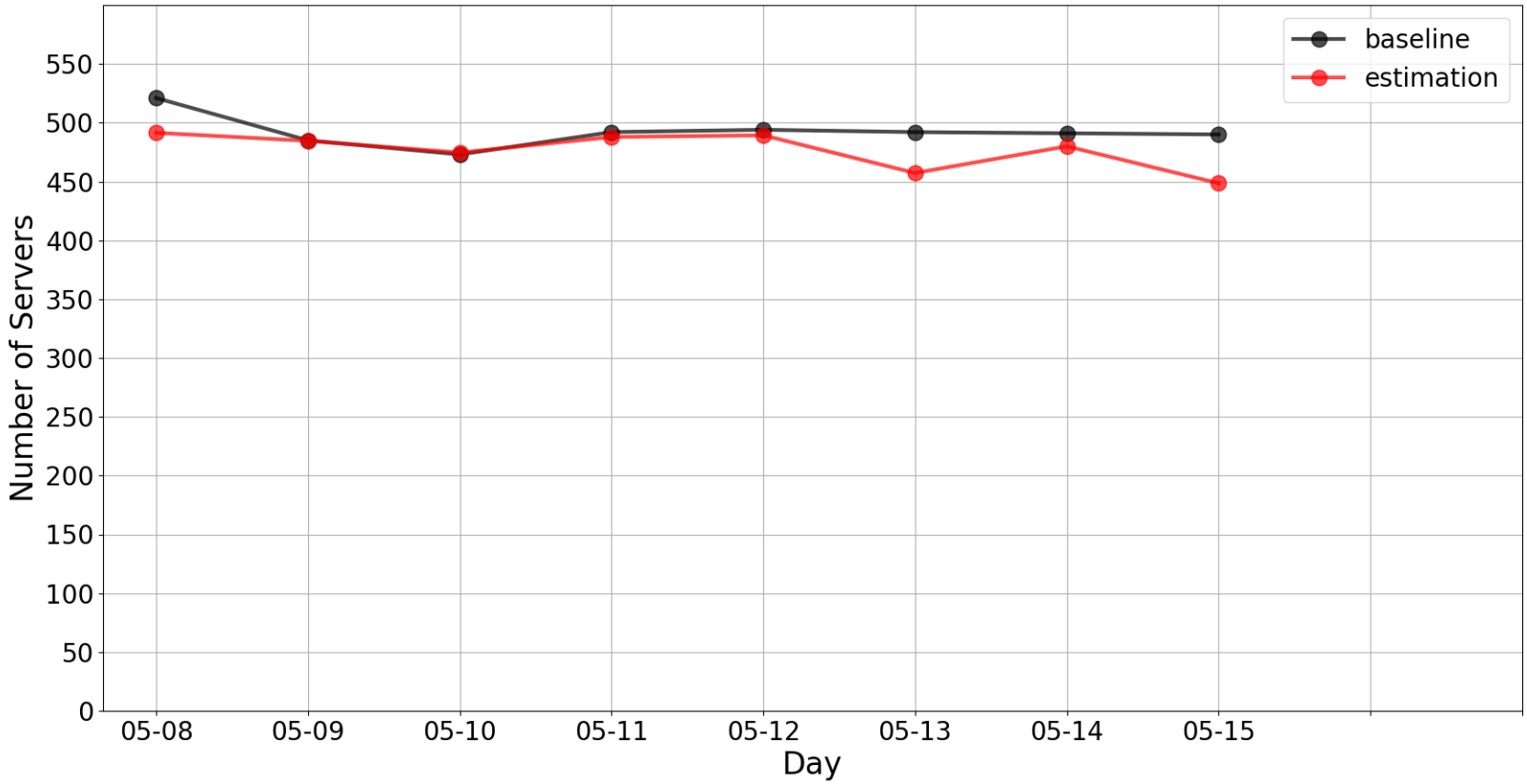
Estimation Result (Orange)



Estimation Result (Green)



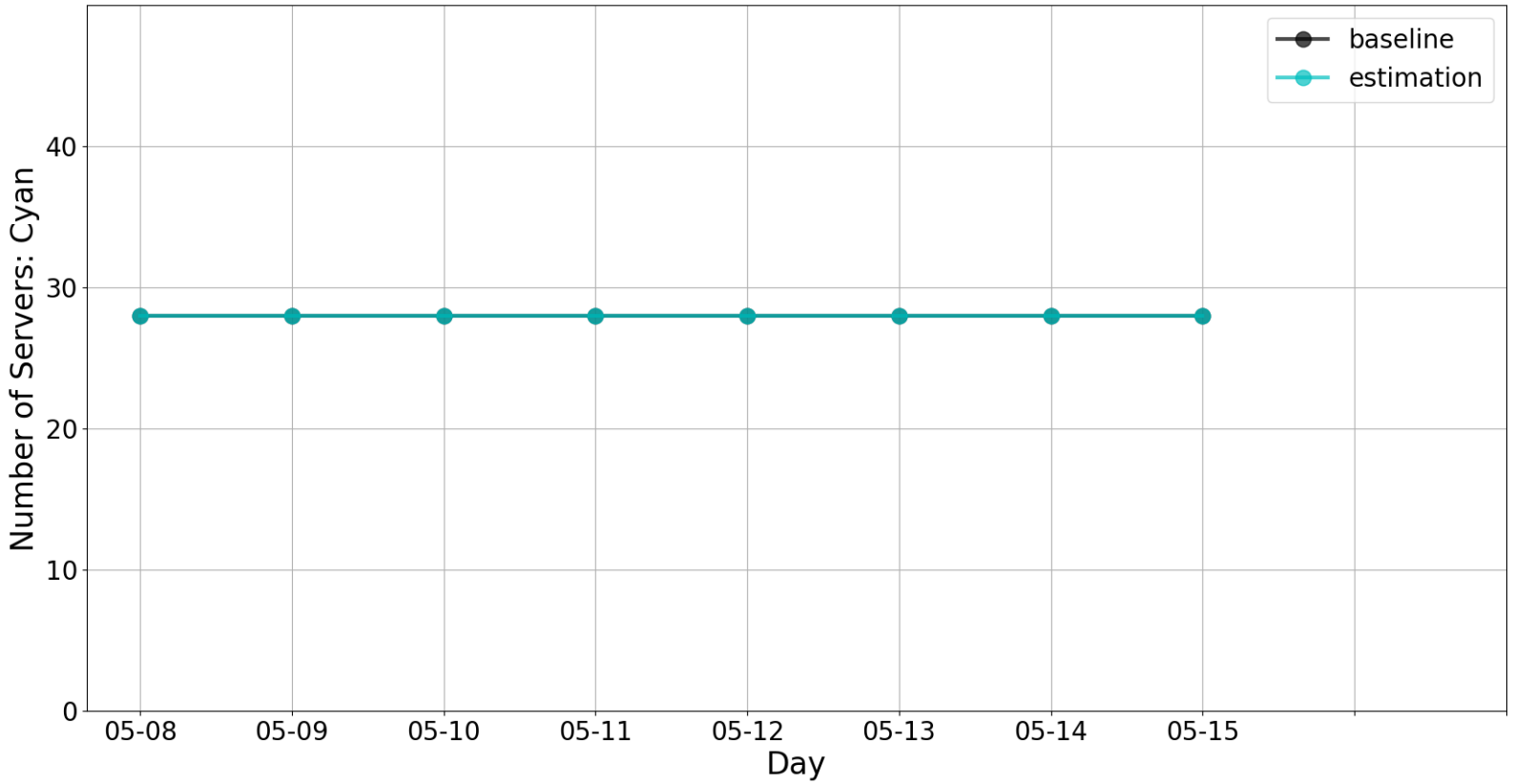
Estimation Result (Total)



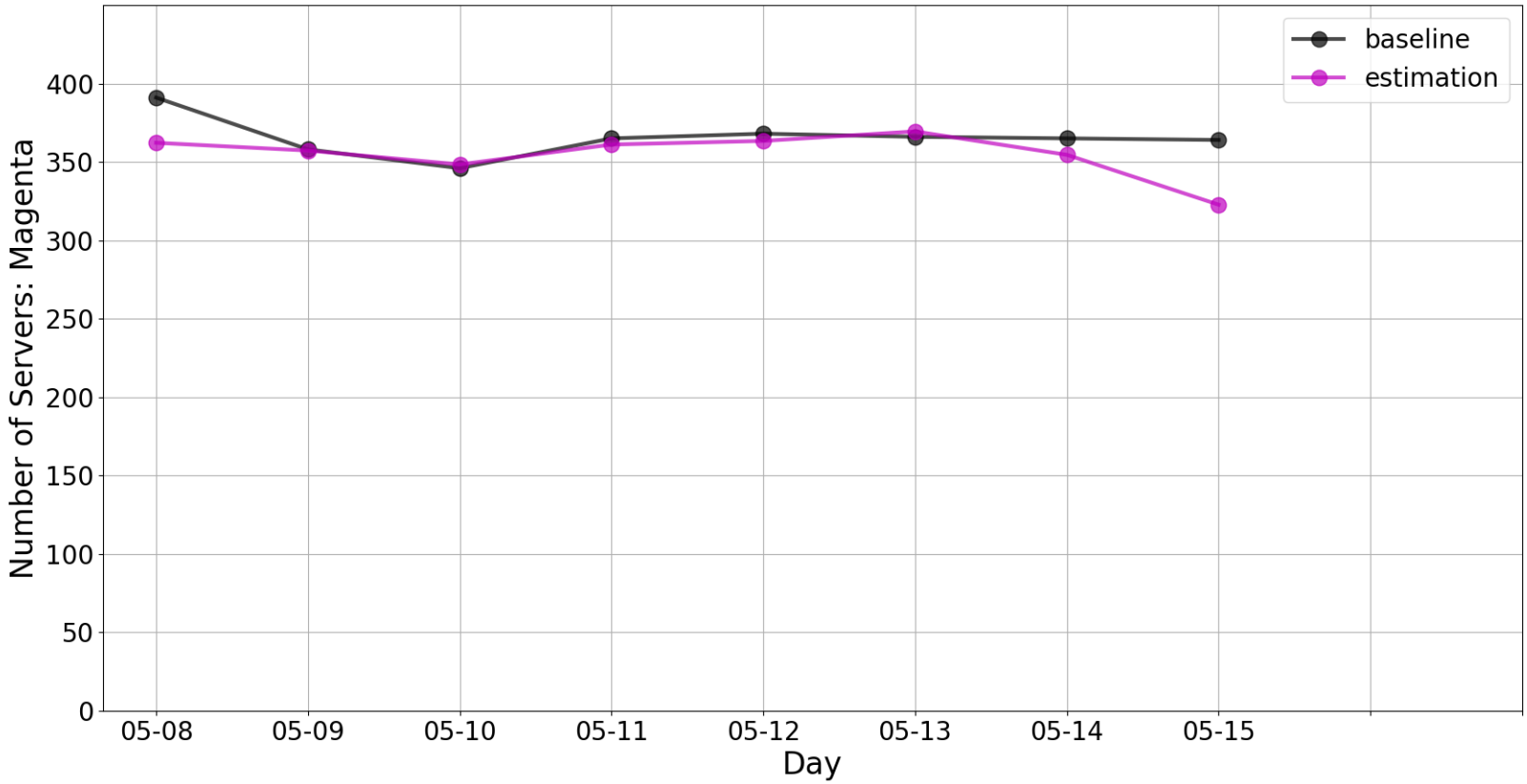
K-means Clustering Phase 2

Group	Server numbers	Average estimation error (%)
Total	538	1.87
Orange	10	16.67
Green	31	16.67
Cyan	28	0
Magenta	408	1.17
Yellow	61	0.13

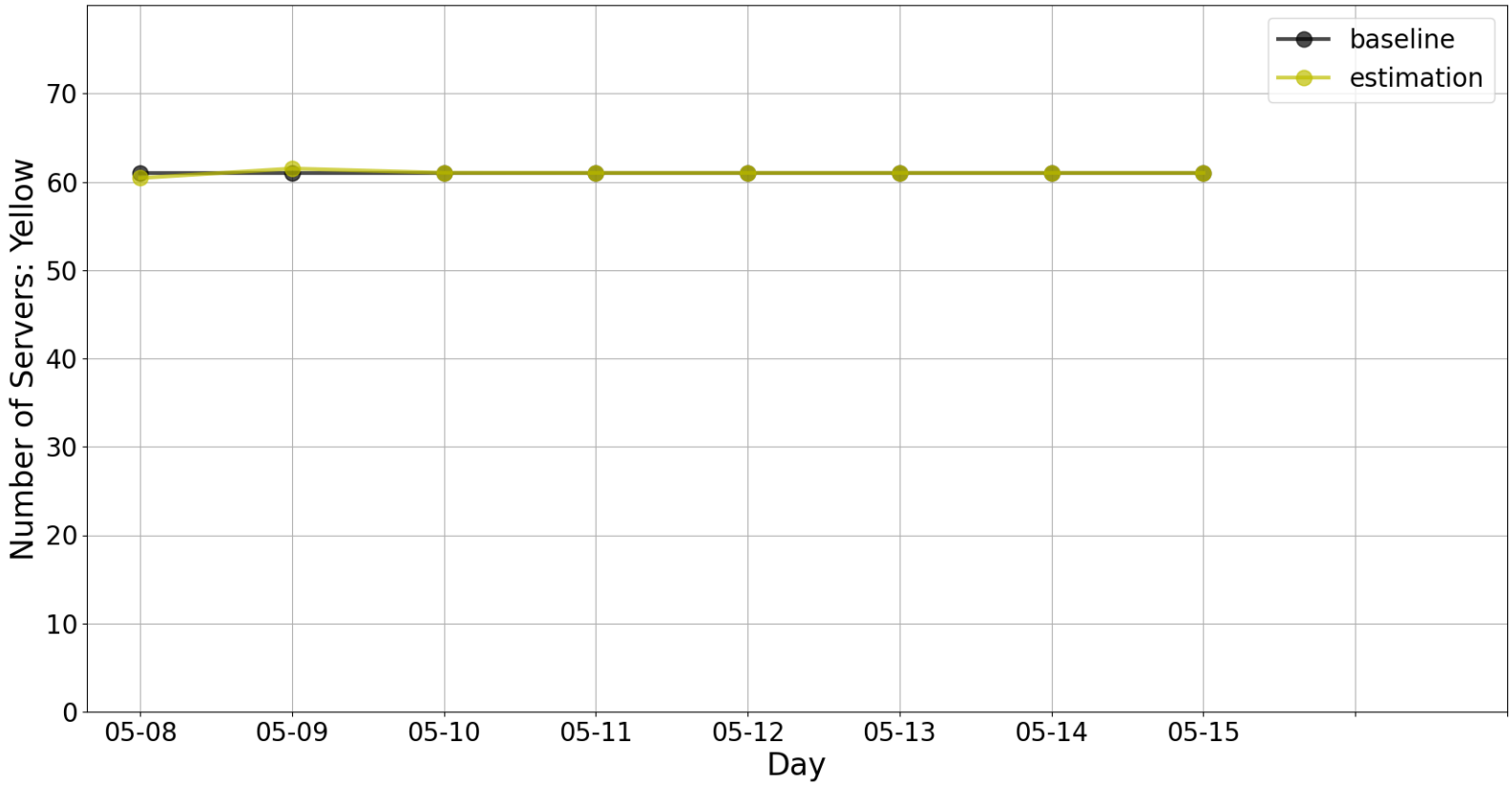
Estimation Result (Cyan)



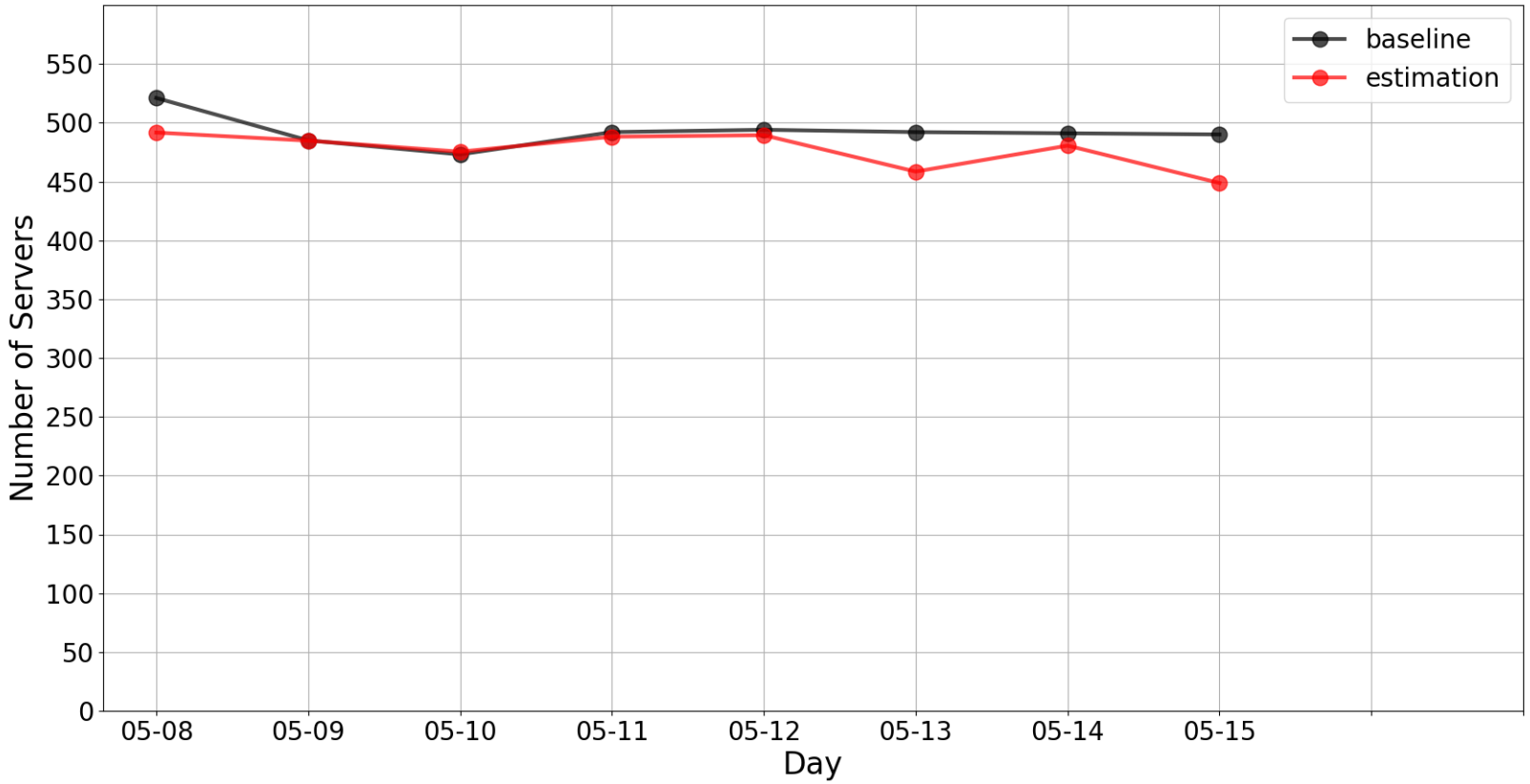
Estimation Result (Magenta)



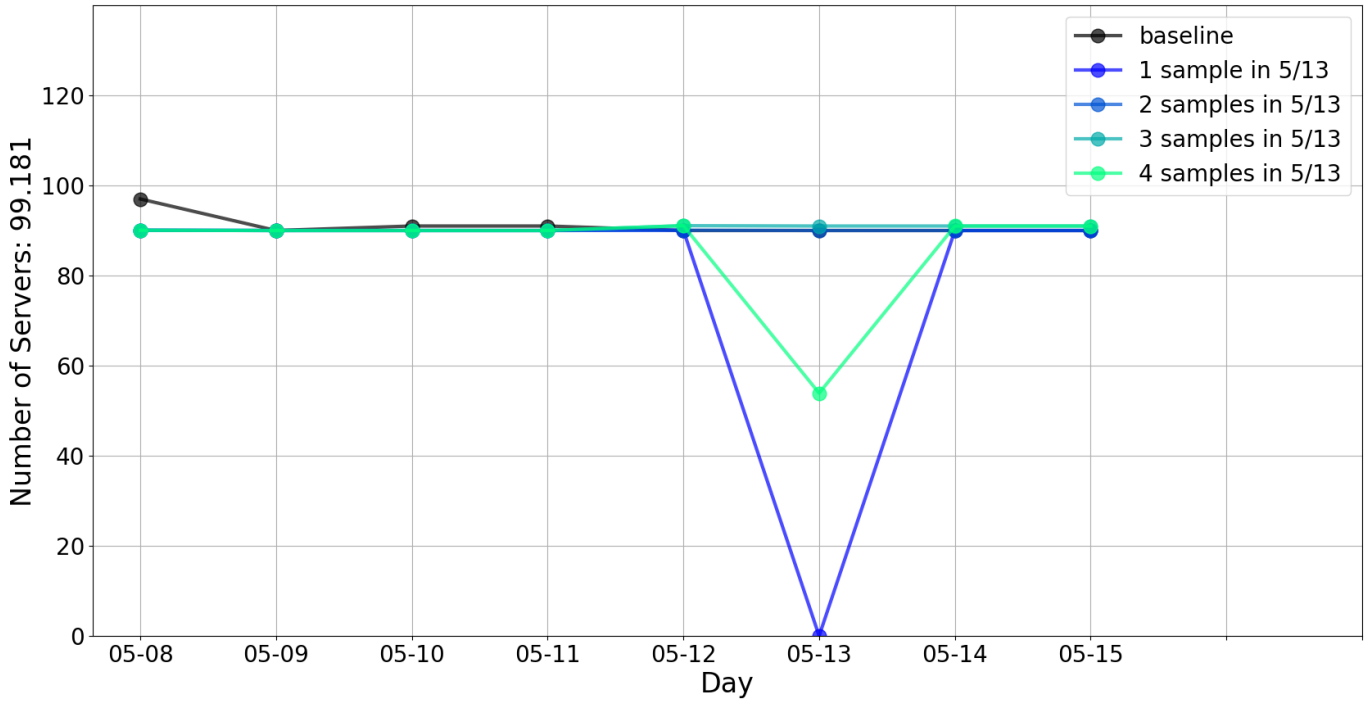
Estimation Result (Yellow)



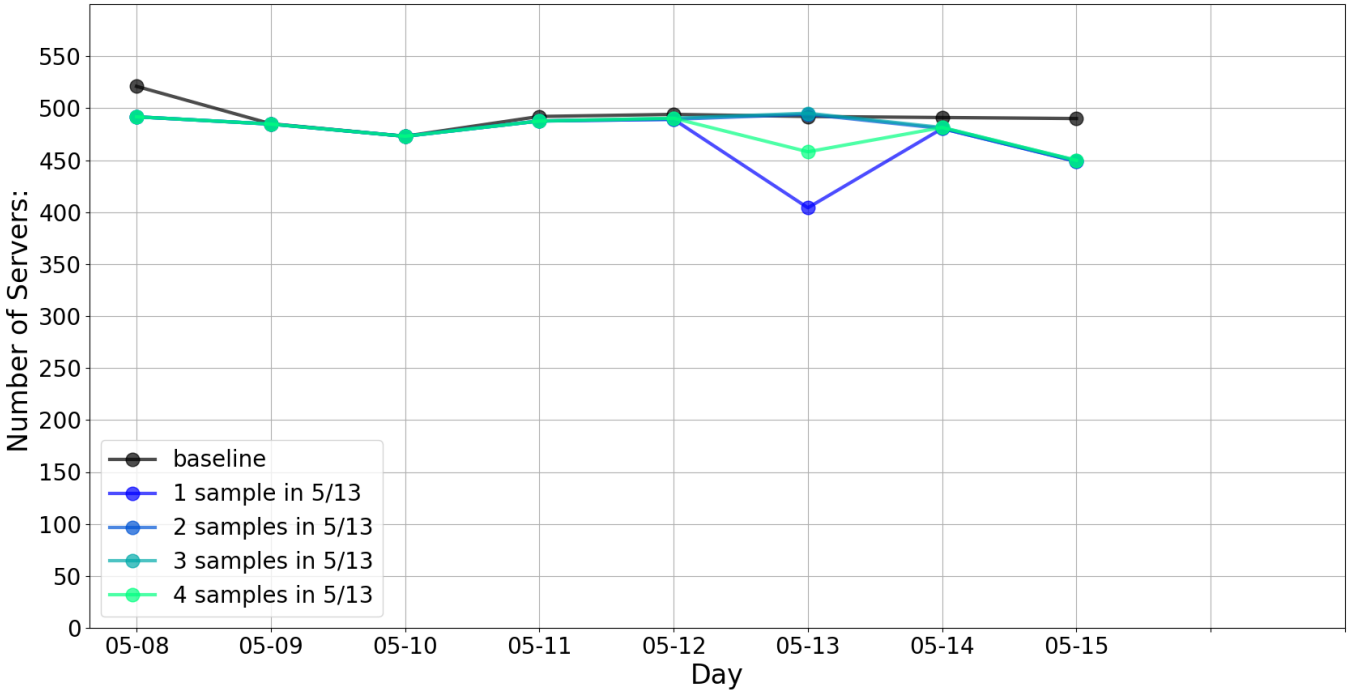
Estimation Result (Total)



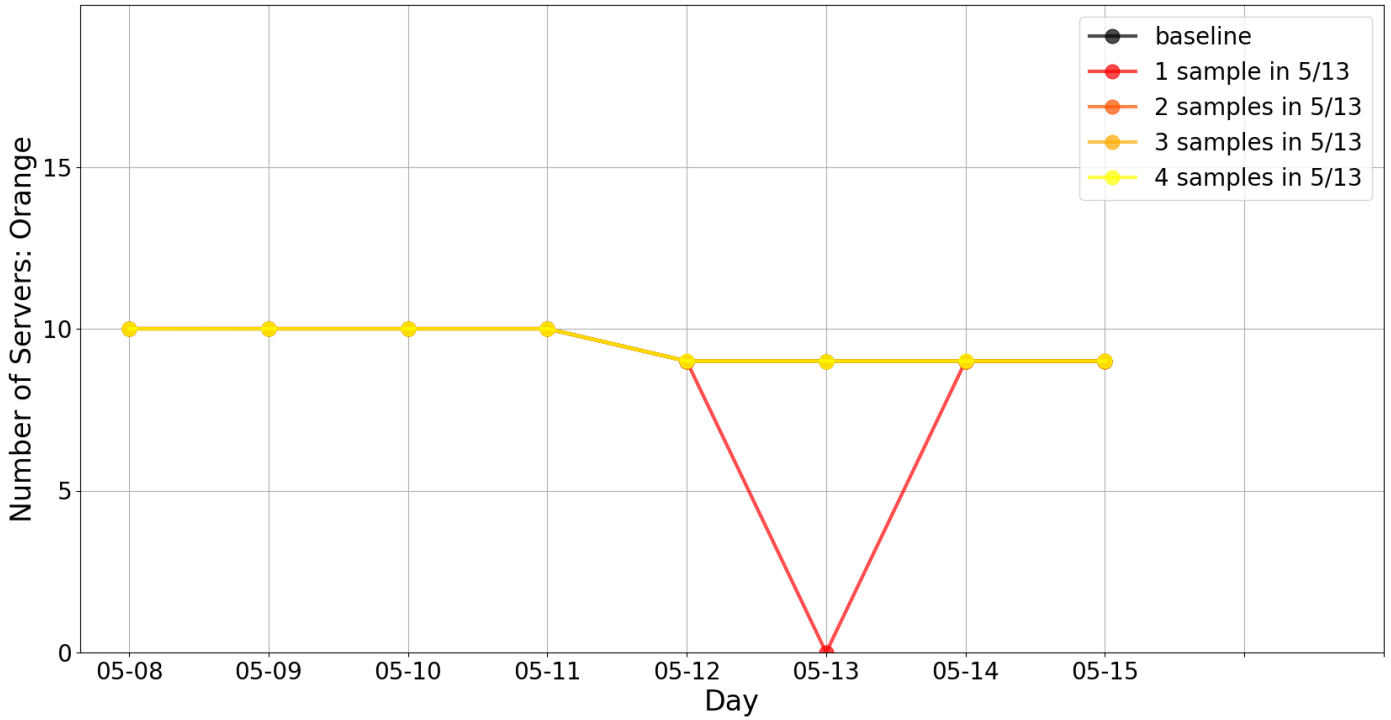
Improve Accuracy with Extra Samples ("99.181")



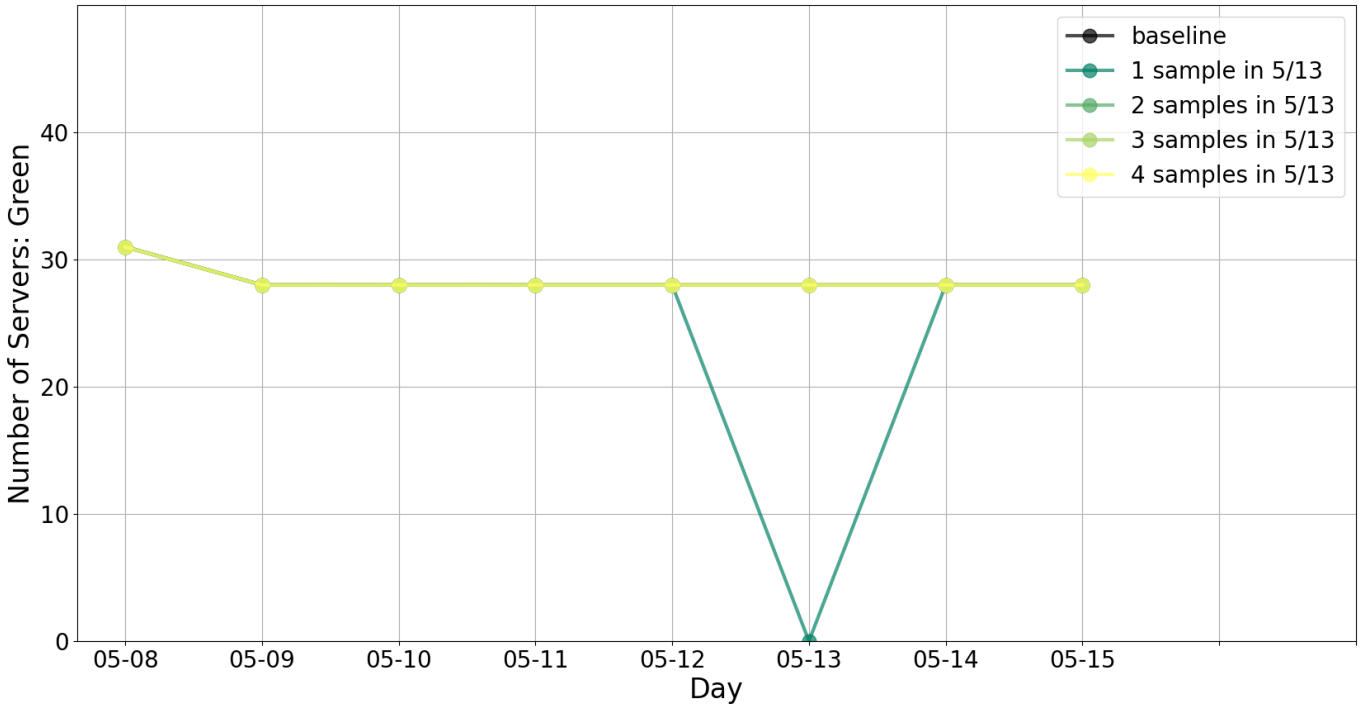
Improve Accuracy with Extra Samples (16-bit IP Prefixes)



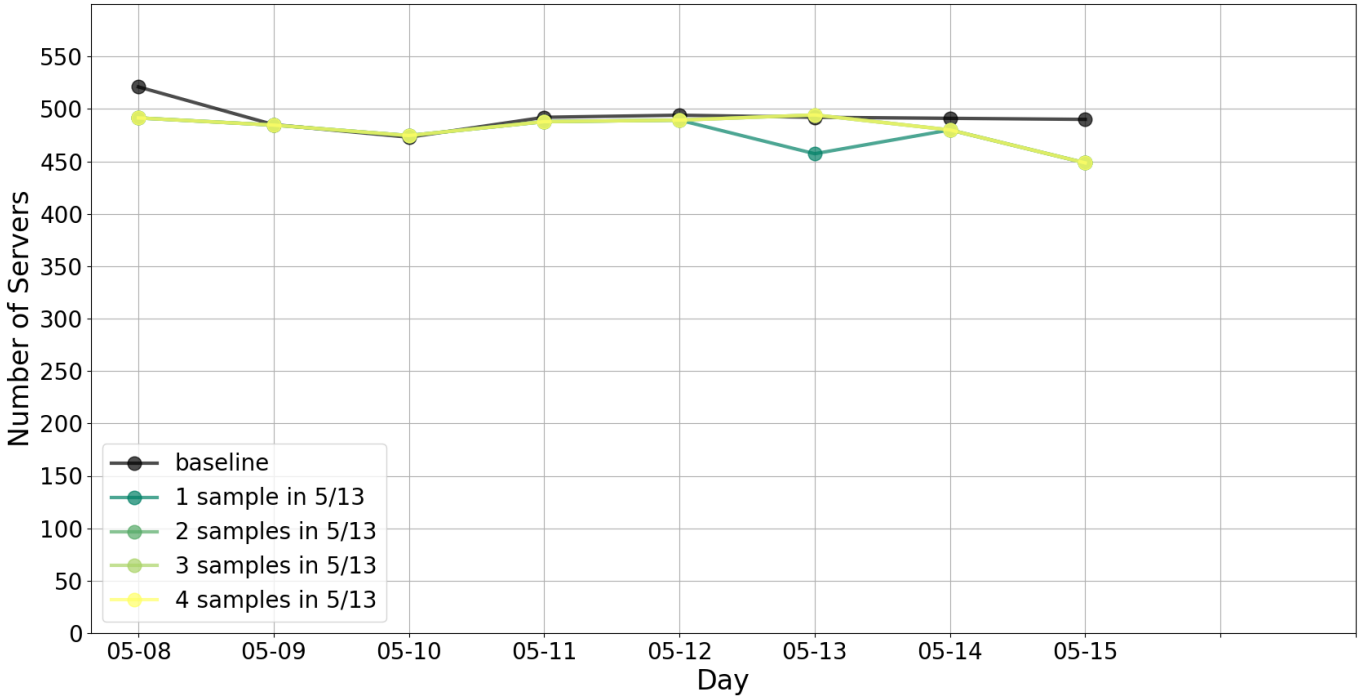
Improve Accuracy with Extra Samples (Orange Group)



Improve Accuracy with Extra Samples (Green Group)



Improve Accuracy with Extra Samples (3 Groups)



Improve Accuracy with Extra Samples (5 Groups)