Intranet Redirect Detector or Pseudo-Random Subdomain Attack?

Chromium's Impact on Root DNS Traffic

Matthew Thomas, Duane Wessels

Verisign

VERISIGN®

Chromium

- A free and open-source software project from Google.

- Used by Google to develop their Chrome browser.

- Used by many other browsers including:
  - Microsoft Edge, Opera, Amazon Silk, and Brave
  - Mobile browsers including Kiwi, Samsung, Kromite, and Ecosia

- Cumulatively Chromium-based browsers have approximately 70% monthly usage share[1]

[1] https://www.w3counter.com/trends

Chromium Omnibox

- **Purpose**: Merge both location and search fields while offering the user some relevant suggestions or results.

- **Problem**: Is the user entering a search term or domain?
  - **Subproblem**: For domain-like strings, can the network (i.e. local DNS) be trusted to differentiate existent vs non-existent domains?

- **Work-around**: Issue a series of probe URL fetches to determine if the network intercepts and redirects requests for non-existent domain names.

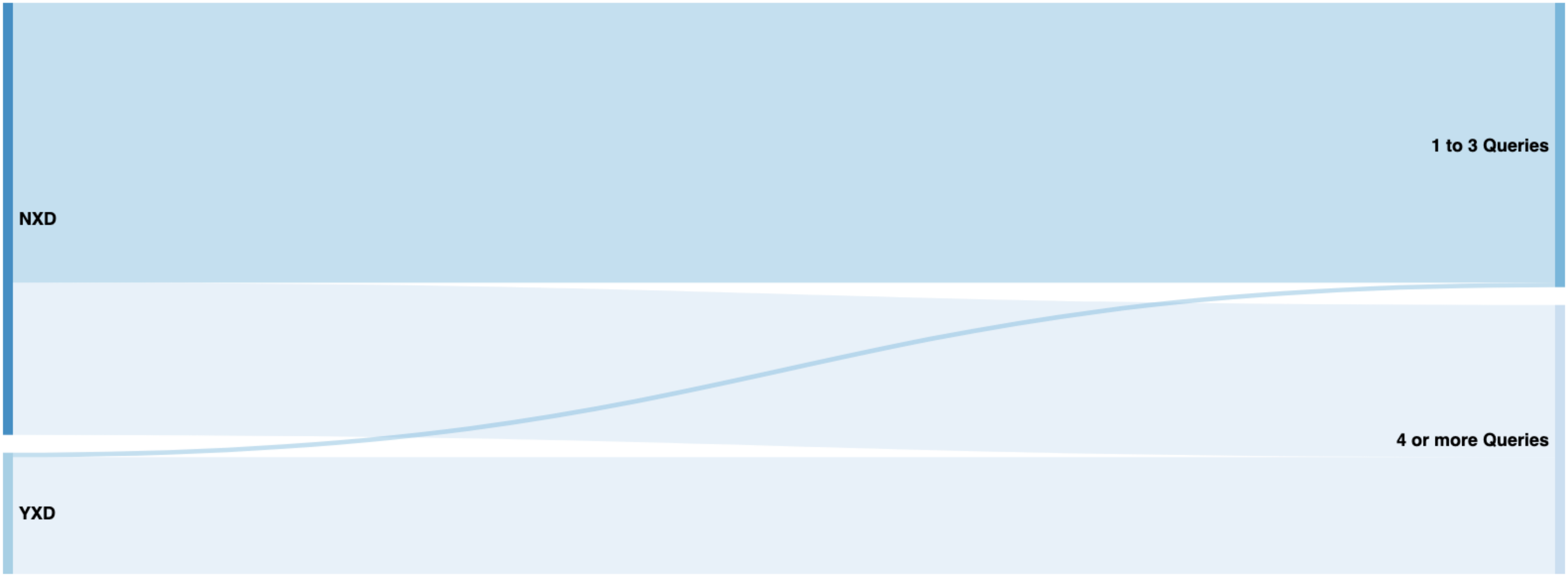
Chromium Probe Design

- Construct three random domain names used in an HTTP request
- Each domain is a random length between 7 and 15 characters
  - Prior to February 2014, Chromium only used 10 character lengths
- Only use characters a-z (case insensitive)
- Due to structure of the names, they should should not exist and the response should be NXDomain
- If any two of the three requests resolve to the same host, that host is stored as the network's "redirect origin"
- This is done at startup, and every IP address and DNS server change

Examples:

http://muthsiengks/	http://nghwirjse/	http://sdhghamsdfe/
http://muthsiengks.corp/	http://nghwirjse.home/	http://sdhghamsdfe.hub/

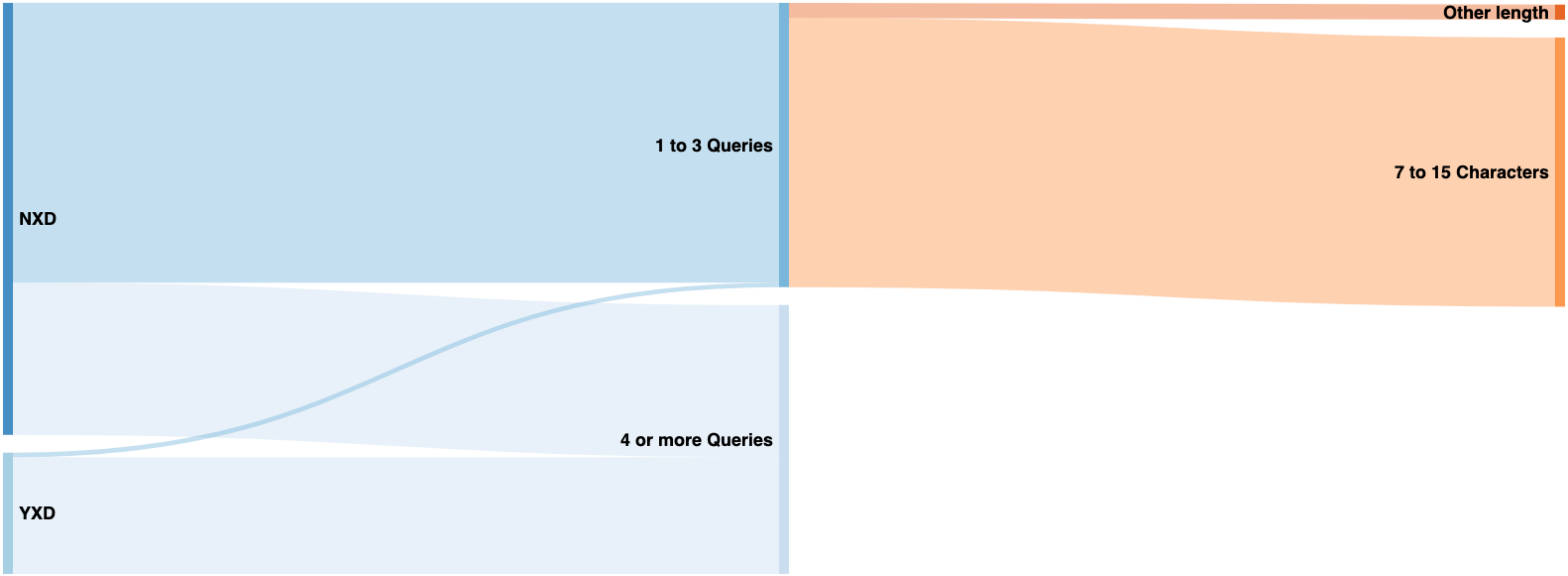
Identifying Chromium Queries



- Number of possible strings generated by Chromium:

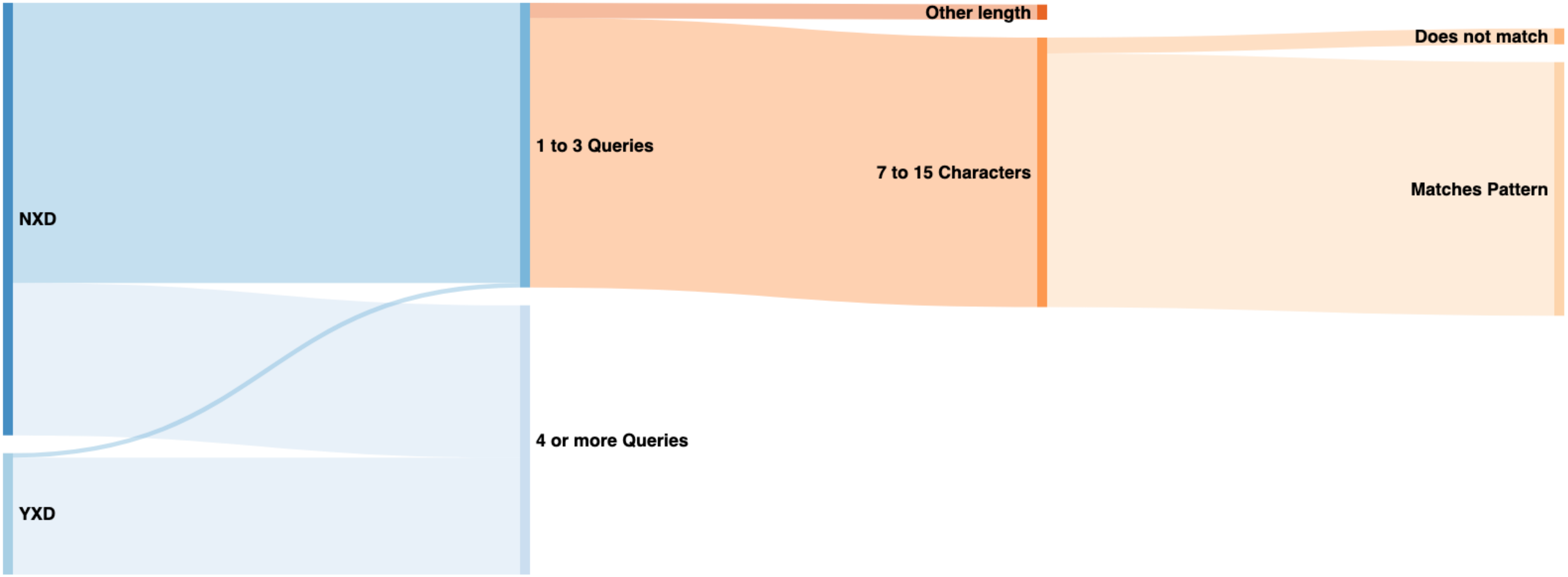
  n^k n=26 and 7 <= k <= 15
- For a given day, we should likely only see that random string a few times at most

Identifying Chromium Queries



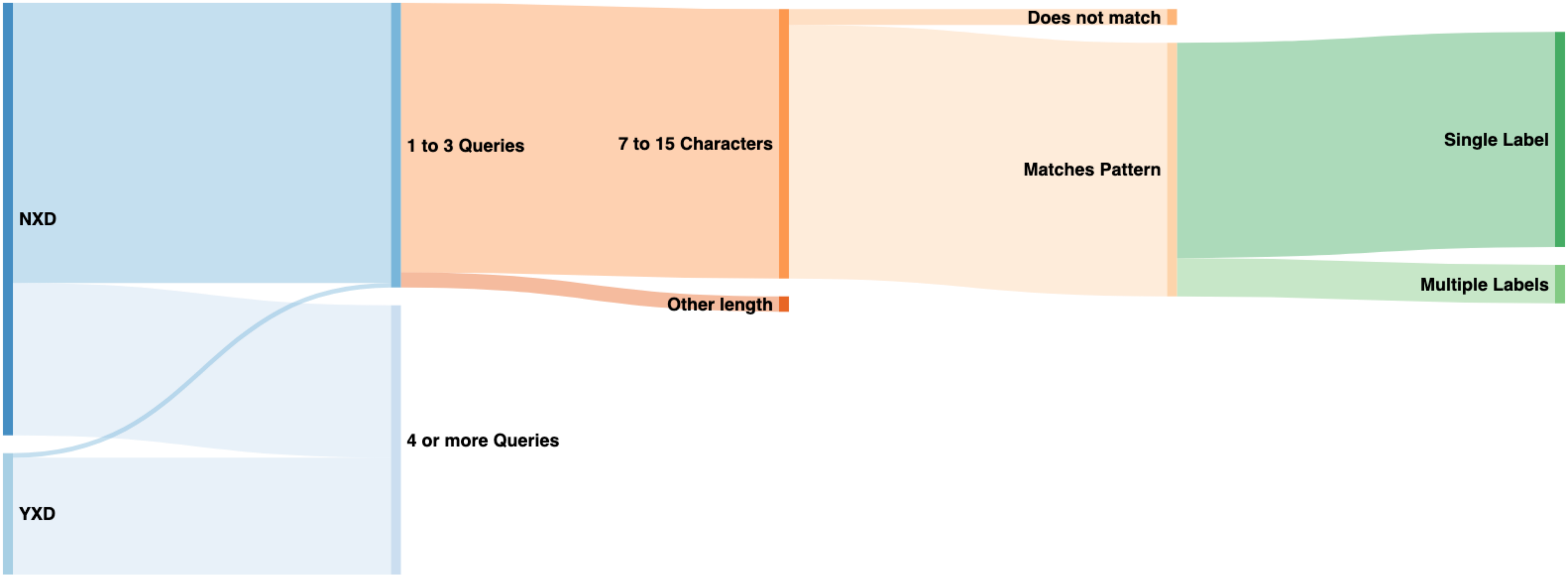
- Leftmost label is between 7 and 15 characters long

Identifying Chromium Queries



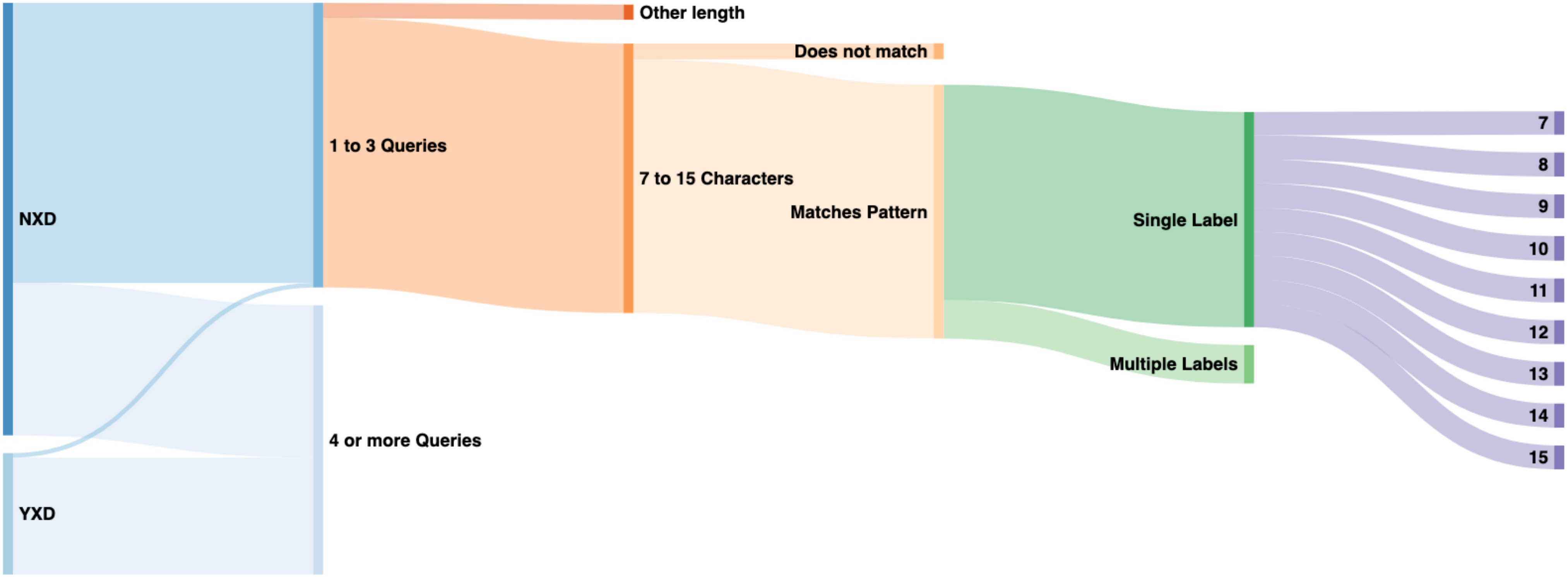
- Leftmost label only contains characters between ‘A’ and ‘Z’ (case-insensitive)

Identifying Chromium Queries



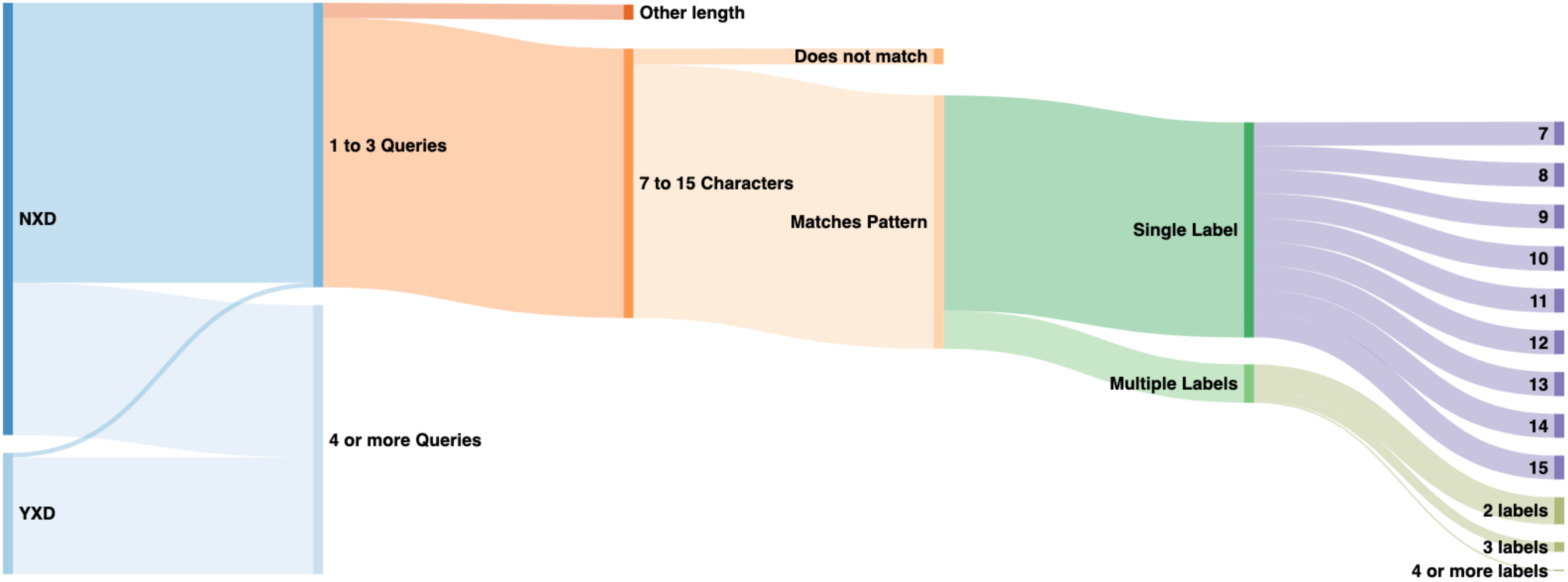
- Is the leftmost label the only label present or are there multiple labels?

Identifying Chromium Queries



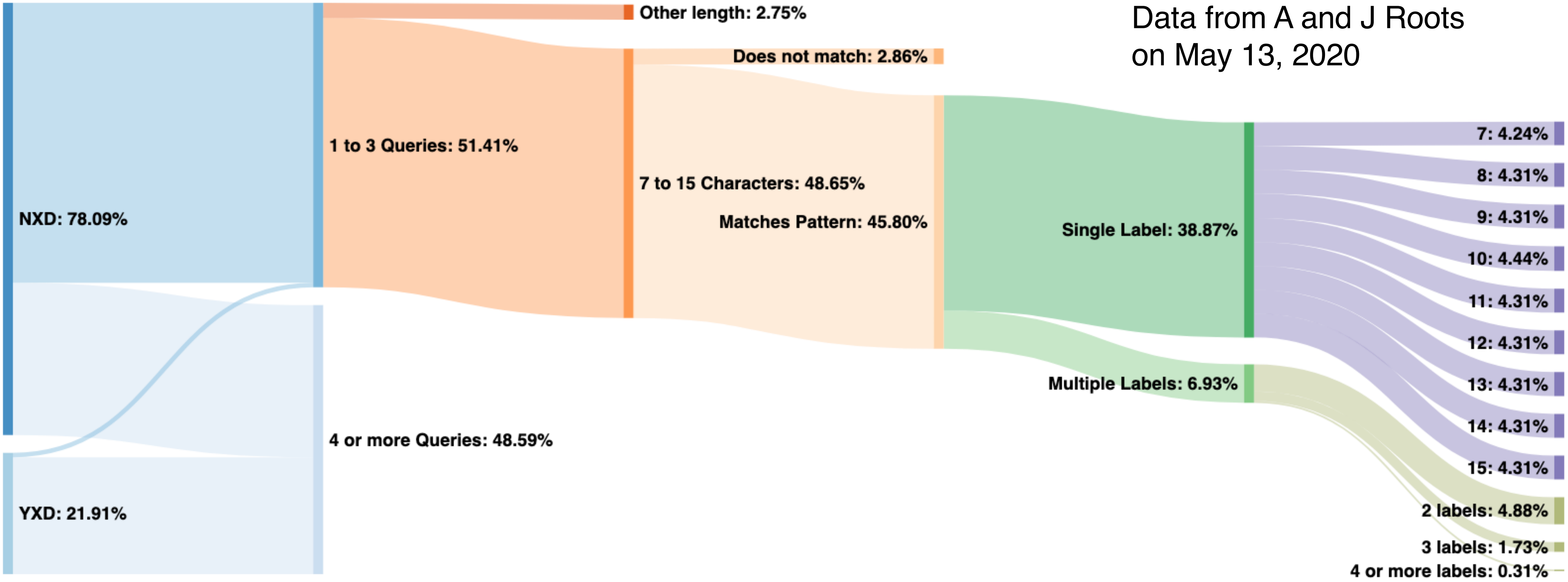
- Record the length of matching strings

Identifying Chromium Queries



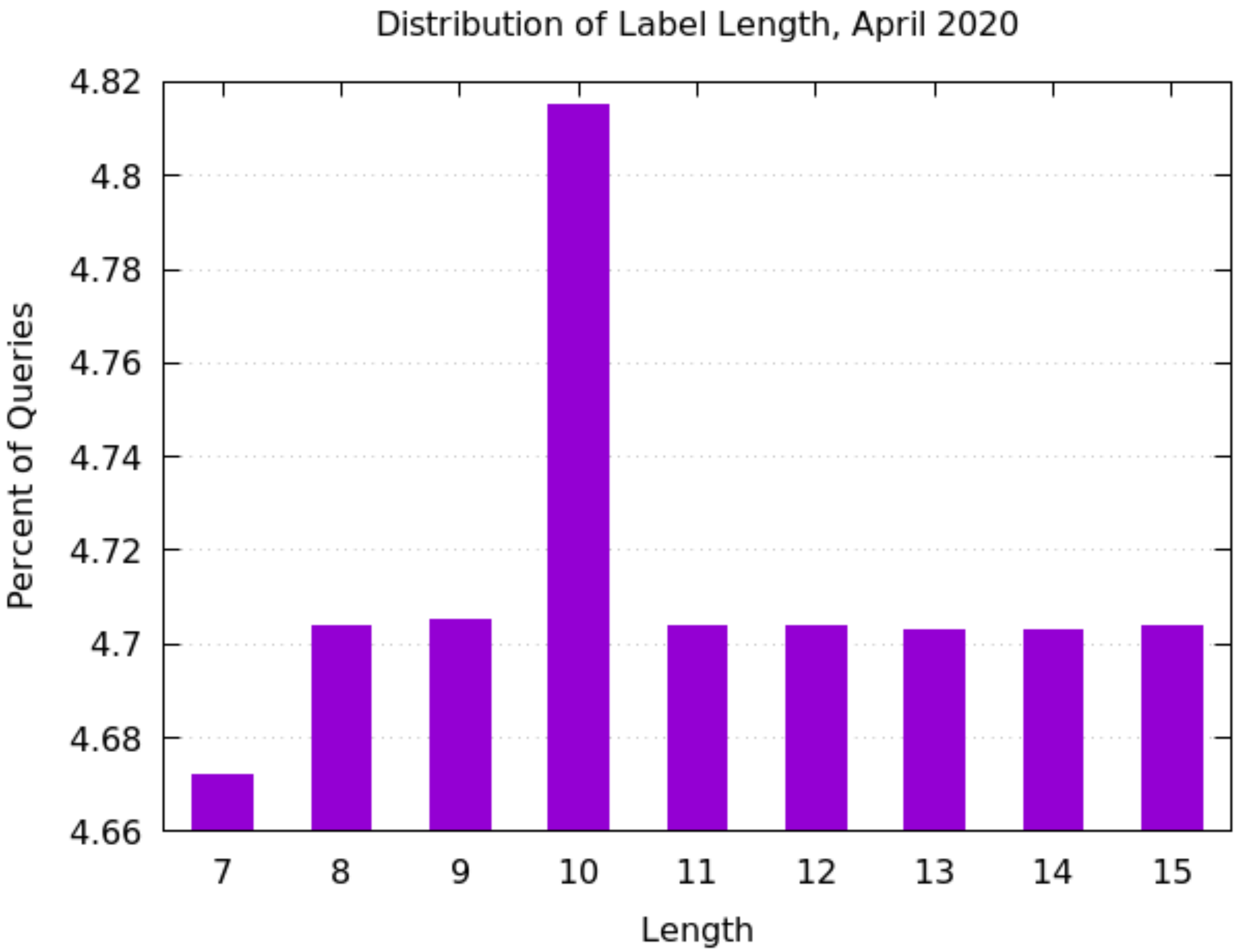
- How many labels are present in multiple-label names?

Identifying Chromium Queries



Data from A and J Roots
on May 13, 2020

- In this data sample, ~45% of DNS queries issued to A and J roots are likely from Chromium (see “Matches Pattern” above)
- Even distribution of query length among single label queries with a slight bump at 10 due to legacy probe algorithm.
- Suffix search list appendage apparent in multiple label queries.

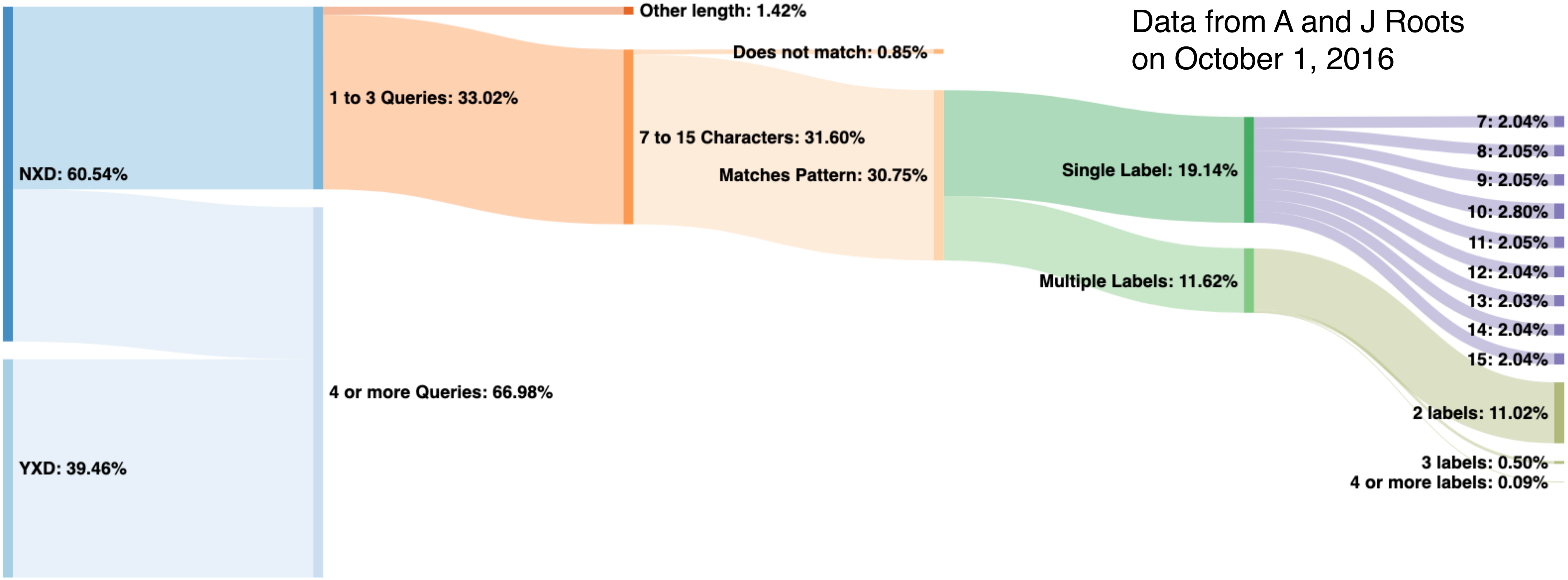

Distribution of Label Length, April 2020
Percent of Queries
4.82
4.8
4.78
4.76
4.74
4.72
4.7
4.68
4.66
7
8
9
10
11
12
13
14
15
Length

False Negatives and Positives

- Some names that match but aren’t random (overcount)
  - duanesblog.unpopular.site
- Some random names in root traffic predating Chromium feature (overcount)
- Retries and aggressive suffix searching lead to extra queries that exceed our threshold (undercount)
- Aggressive NSEC caching (undercount)
- Geoff Huston said that DNS has a long memory[2]; some queries may be replayed (both undercount and overcount)

[2] https://www.potaroo.net/ispcol/2019-02/nxd.html

October 2016 – Botnet traffic



Data from A and J Roots
on October 1, 2016

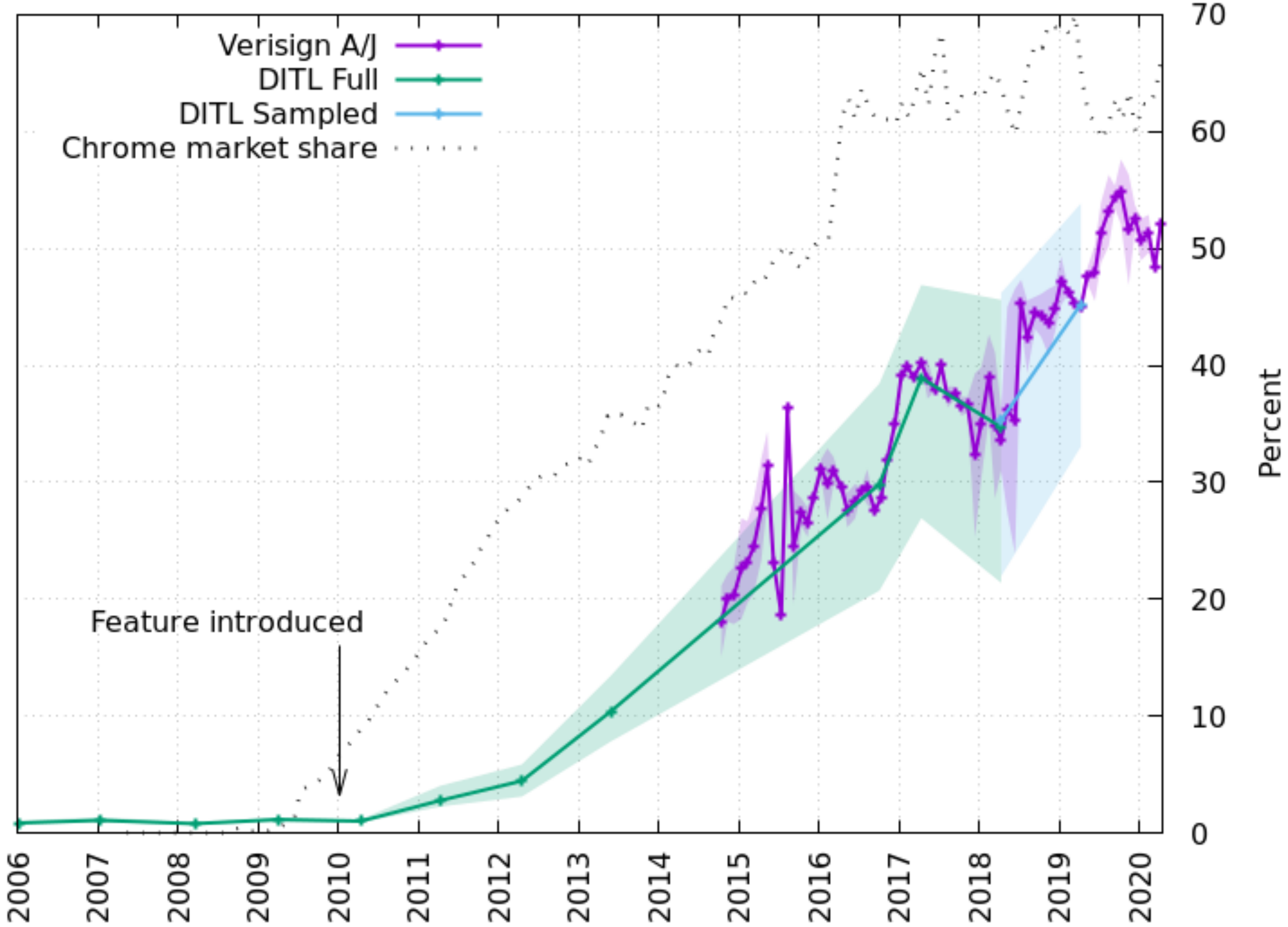
- Botnet sent pseudo-random subdomain queries ending in “.null”

Longitudinal Measurement of Chromium Queries

- DITL data from 2006 - 2019
  - Gap from 2014 - 2016 due to unavailability of an OARC file server
  - Used October 2016 data from root ZSK length change
  - 2017 - 2019 data very difficult to analyze properly given its size and constraints of analysis server
  - Percentages vary significantly among root server identities
- Verisign data for a.root-servers.net and j.root-servers.net
  - 2015 - 2020
  - NXDomain only 2015 - 2016, adjusted using RSSAC002 data[3]

[3] https://github.com/rssac-caucus/RSSAC002-data

Queries Matching Chromium Probe Pattern as Percent of Total Root Traffic



Open Questions

- Is the load placed on the root servers for determining NXDomain interception proportional to the problem case Chromium is attempting to solve?
  - What criteria were used to determine that 3 queries is optimal?
  - Would this work just as well under a reserved TLD?
  - How common is NXDomain interception?
    - Observed in about 1% of RIPE Atlas probes
- To what extent do suffix search lists make this even worse?
- What technical solutions can help mitigate root pollution?
  - Are approaches used by Firefox more appropriate?
  - Aggressive NSEC Caching[1], QnameMin[2], and NXD Cut[3]?

[1] https://tools.ietf.org/html/rfc8198
[2] https://tools.ietf.org/html/rfc7816
[3] https://tools.ietf.org/html/rfc8020

powered by

VERISIGN®