# Around the Web in Six Weeks: Documenting a Large-Scale Crawl

Sarker Tanzir Ahmed, Clint Sparkman, Hsin-Tsang Lee, and Dmitri Loguinov

Internet Research Lab
Department of Computer Science and Engineering
Texas A&M University

April 29, 2015

# Agenda

- Introduction
- Background
- Crawl Analysis
  - Page-Level
  - Server-Level
- Internet Coverage
- Extrapolation
- Conclusion

# Introduction

- Web crawling is a challenging experiment
  - Its perceived difficulty hinders non-commercial endeavors
- Industry has been the major player
  - Reluctant to disclose actual methodology
- Academic endeavors are limited
  - Popular belief that a Internet-wide requires huge hardware setup
  - Most published crawls are rather limited in size and span in the Internet and lack useful details about the crawl
  - No standard methodology to compare different crawls

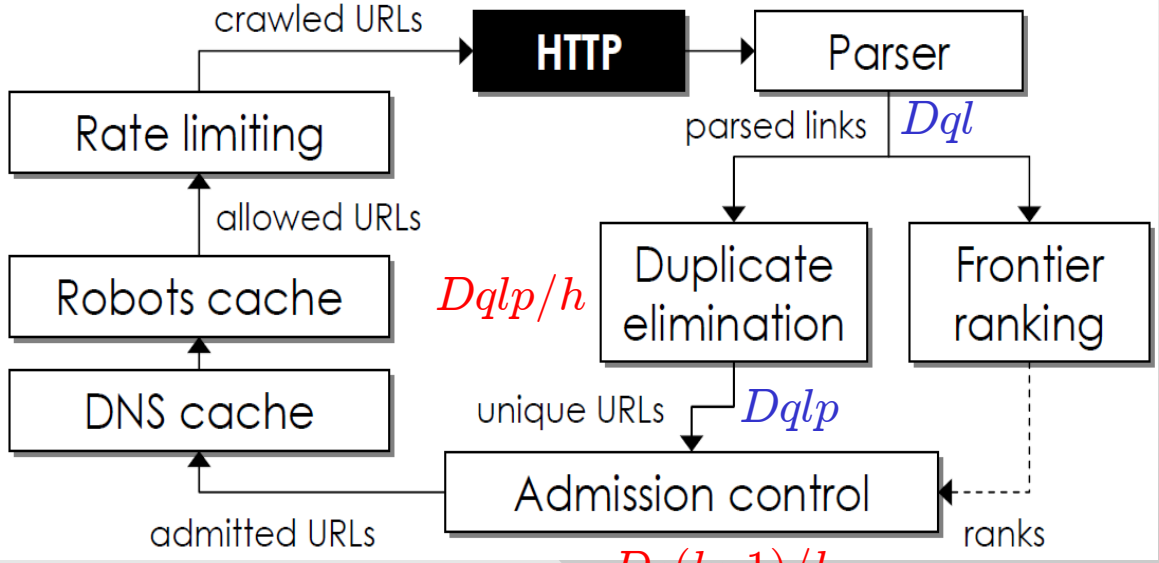
# Introduction (2)

- Our IRLbot crawl experiment in 2007 is the largest non-commercial crawl of the Internet to this date
  - Collected 7.3B pages in 41 days using a single crawler node
- Here the objective is to dissect the collected data
  - Analyze Internet wide coverage, spam avoidance etc
  - Compare to commercial search engines using a novel method

# Agenda

- Introduction
- Background
- Crawl Analysis
  - Page-Level
  - Server-Level
- Internet Coverage
- Extrapolation
- Conclusion

# Background - Inside a Web Crawler




| Term | Description |
|---|---|
| $\mathcal{D}$ | # of downloaded pages |
| $q$ | Fraction of HTML pages |
| $l$ | Links/page |
| $p$ | Fraction of unseen links |
| $h$ | # of crawler nodes |
| $\mathcal{S}$ | Crawl rate (pages/sec) |

- Forms a cycle where each component has to keep up to persist the crawl rate $\mathcal{S}$
- Example: IRLbot's duplicate elimination rate was over 100K/s with peak rate $\mathcal{S}$=3K pps, $m$=$h$=1

# Background – Crawler Design (2)

- Crawl design boils down to a trade-off $\{\mathcal{D}/h, \mathcal{S}/h, q, l\}$
  - Increase in one typically results in decrease in others
- Different methods of scaling $\mathcal{S}$ in existing literature
  - Clear trade-off between $\mathcal{D}$ and $\mathcal{S}$
  - Reduce $q$ by crawling non-HTMLs (Mercator)
  - Eliminate dynamic URLs to reduce $l$ (ClueWeb09)
  - Eliminate disk-based duplicate elimination by RAM-based method (UbiCrawler, WebBase), or by revisiting same pages (Internet Archive)
- None of at-least-50M page crawls have real-time spam avoidance or global frontier prioritization
  - IRLbot uses real-time frontier prioritization

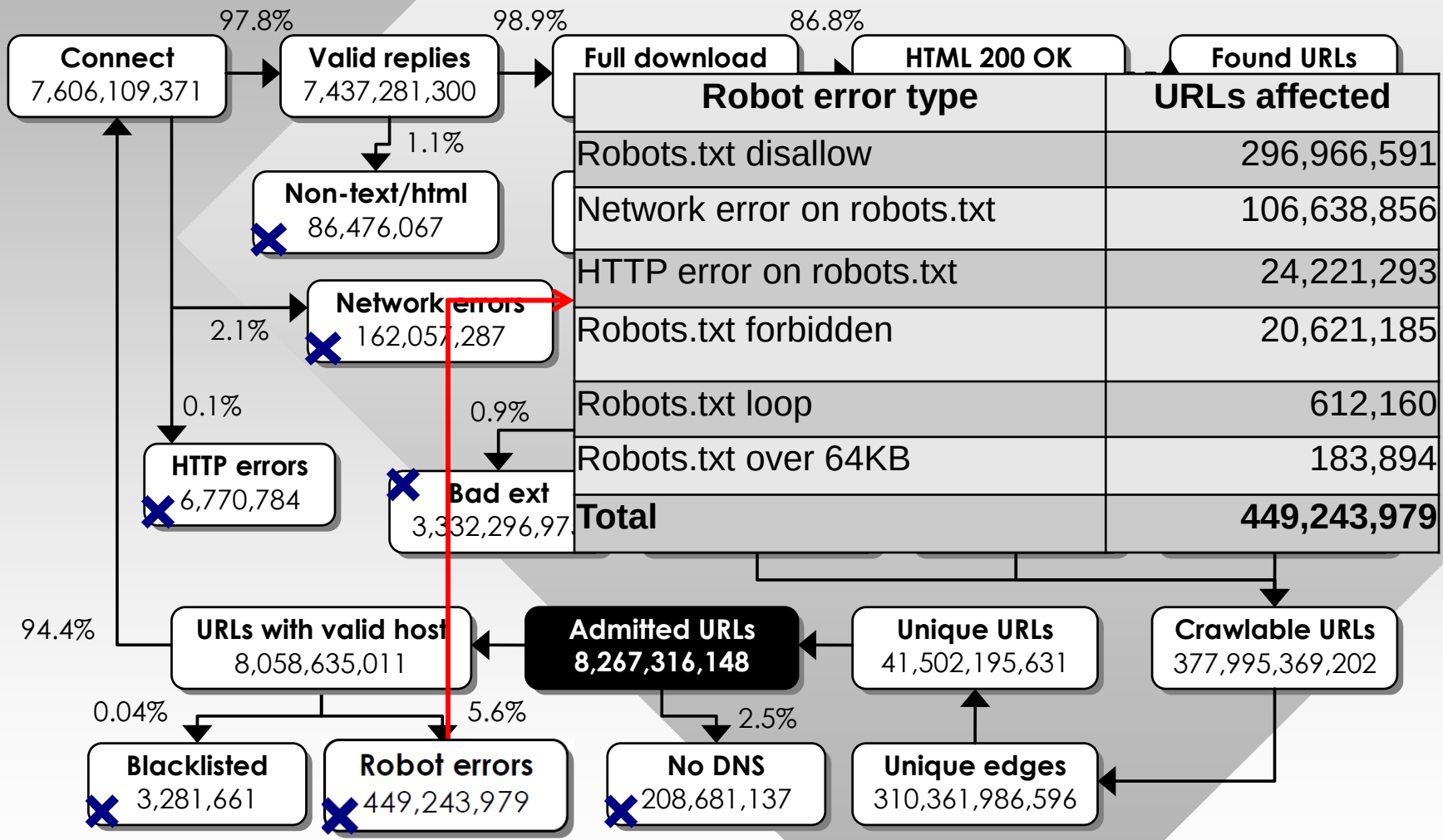
# Background – Crawler Design (3)

- Among distributed crawlers, one of the most prominent is ClueWeb09
  - Parallelized Apache Nutch to 1600 processors in Google-NSF-IBM cluster and discarded all dynamic links (i.e., dropping $l$ by $84\%$)
  - Crawled 1B pages in 52 days at average rate $222$ pps
- Some IRLbot Configuration and Features
  - Used $m{=}h{=}1$, (i.e., one single crawler node, seeded from only $\mathrm{www.tamu.edu}$)
  - Highest $q$ and unrestricted $l$
  - Used real-time frontier prioritization based on the PLD graph
  - Rate $\mathcal{S}$ and $\mathcal{D}$ determined by factors outside our control (i.e., university bandwidth)
  - Collected $\mathcal{D}{=}7.3\mathrm{B}$ pages in $41$ days at average rate $2\mathrm{K}$ pps

# Agenda

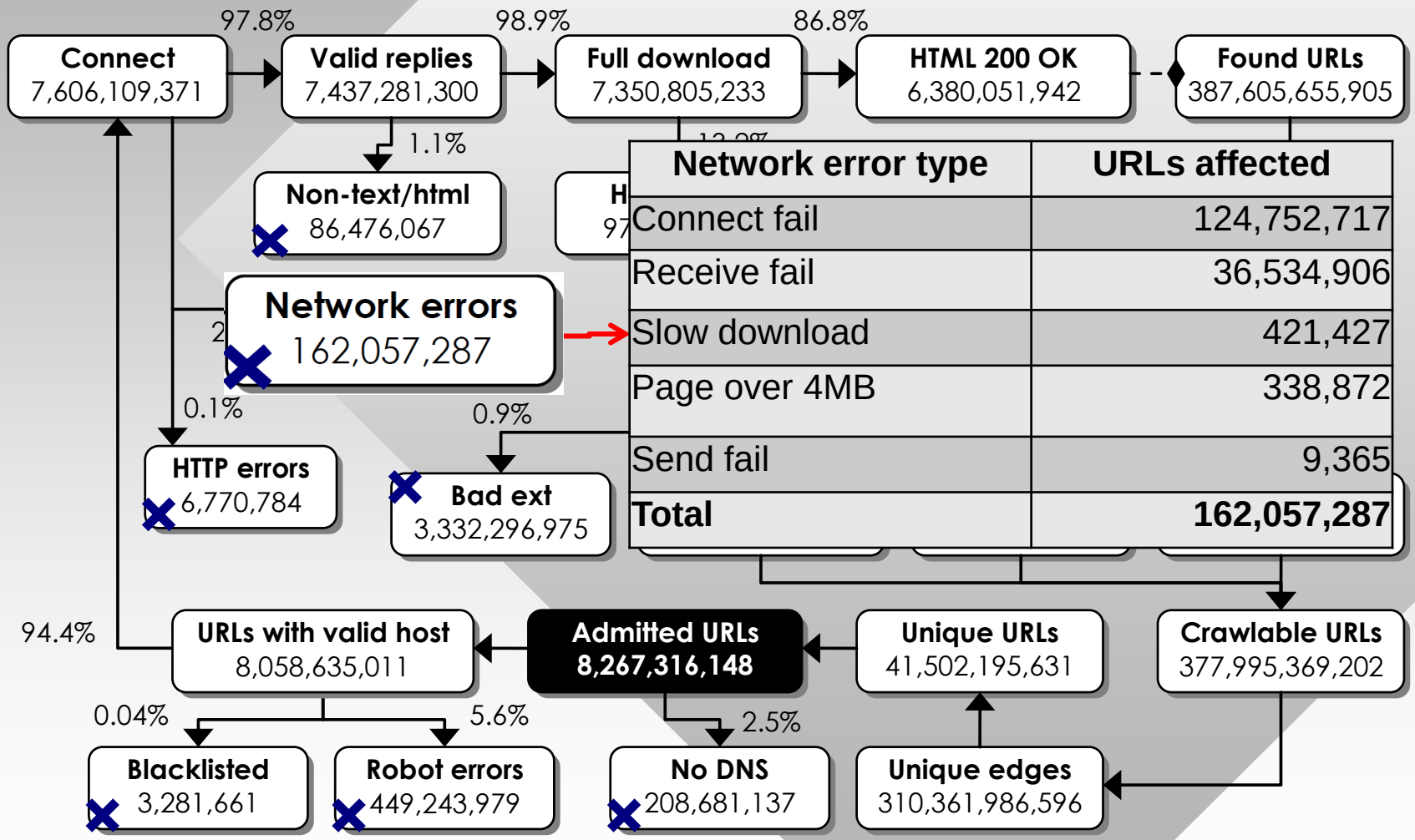
- Introduction
- Background
- Crawl Analysis
  - Page-Level
  - Server-Level
- Internet Coverage
- Extrapolation
- Conclusion

# Page-Level Analysis – URL Cycle

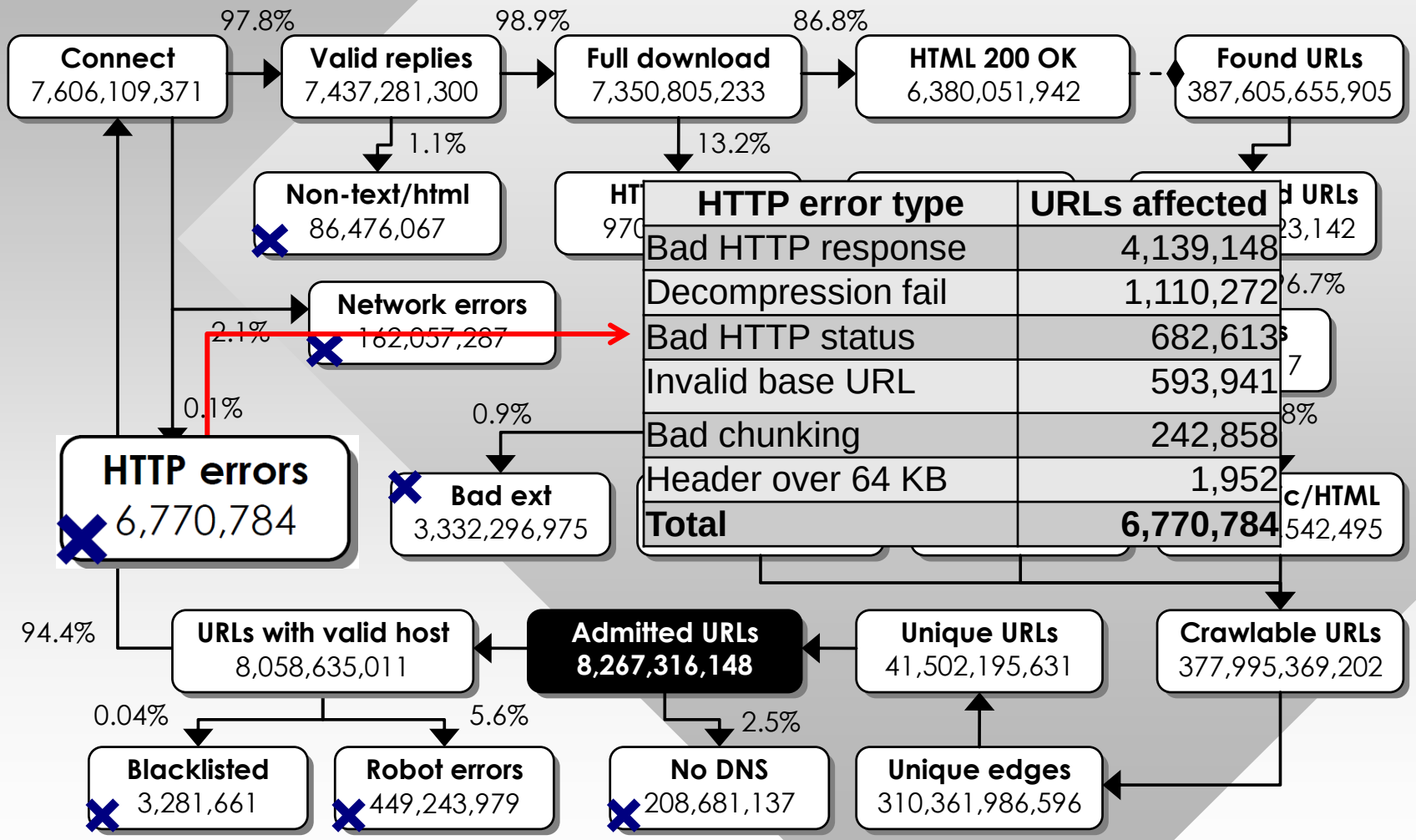



| Robot error type | URLs affected |
|---|---|
| Robots.txt disallow | 296,966,591 |
| Network error on robots.txt | 106,638,856 |
| HTTP error on robots.txt | 24,221,293 |
| Robots.txt forbidden | 20,621,185 |
| Robots.txt loop | 612,160 |
| Robots.txt over 64KB | 183,894 |
| **Total** | **449,243,979** |

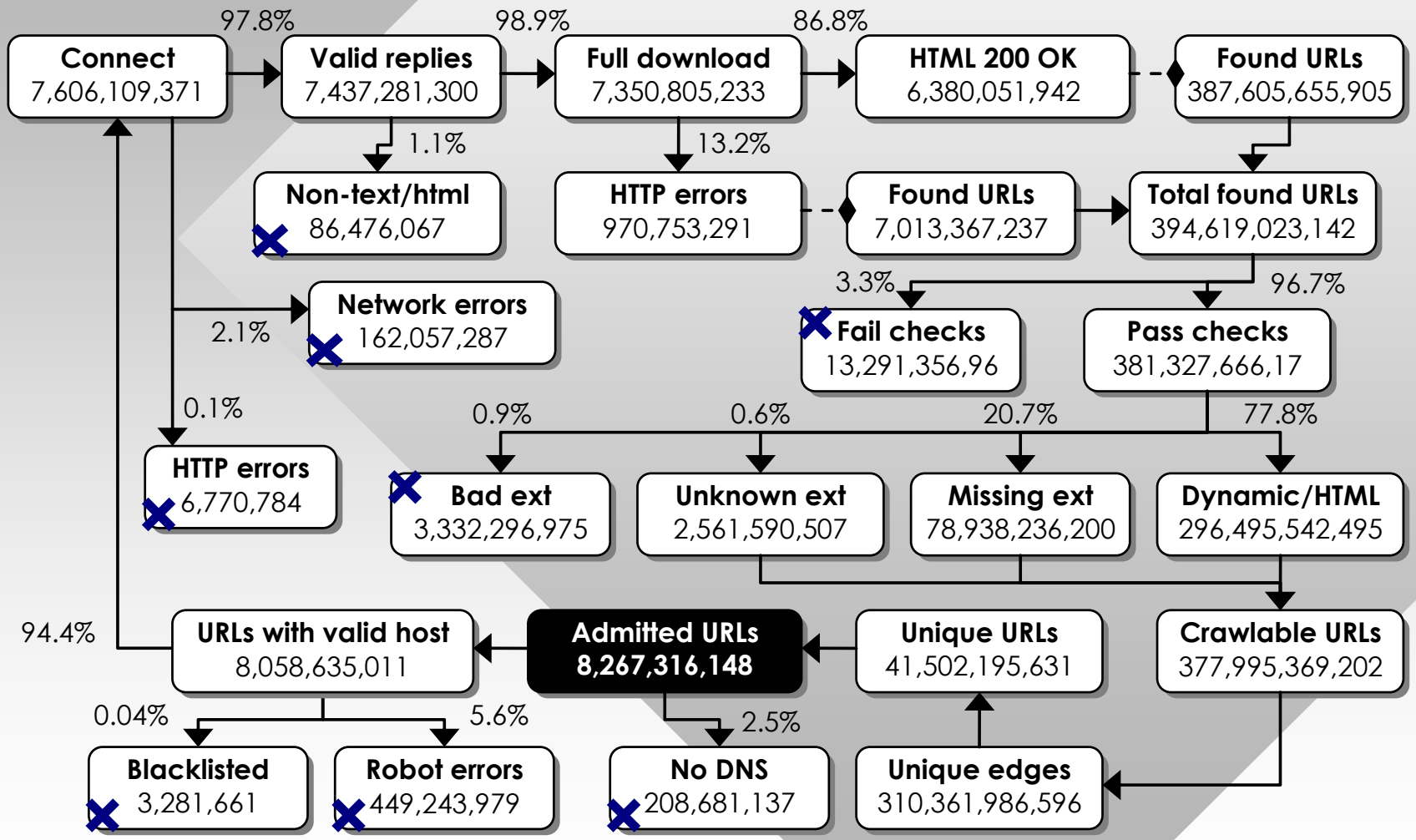
# Page-Level Analysis – URL Statistics



# Page-Level Analysis – URL Statistics



# Page-Level Analysis – URL Statistics



# Page-Level Analysis - A Few Notes

- Countering Spam
  - Did real-time PLD ranking on the current web graph
  - Treated $301/302$ as regular links (processed through cycle)
  - Detected slow downloads (no data for $60$ sec or takes more than $180$ sec)
  - Detected infinite data stuffing and cut off after $4$ MB
- Avoid non-HTMLs
  - Only processed pages with "Content-type: text/html" ($86.5$M discarded objects would take $346$ TB in the worst case)
  - Transmitted "Accept: text/html" header field, but resulted in only $6.6\%$ reduction, while extension filtering leads to $0.37\%$ (not very effective!)
- The result is $8.3$ KB per object

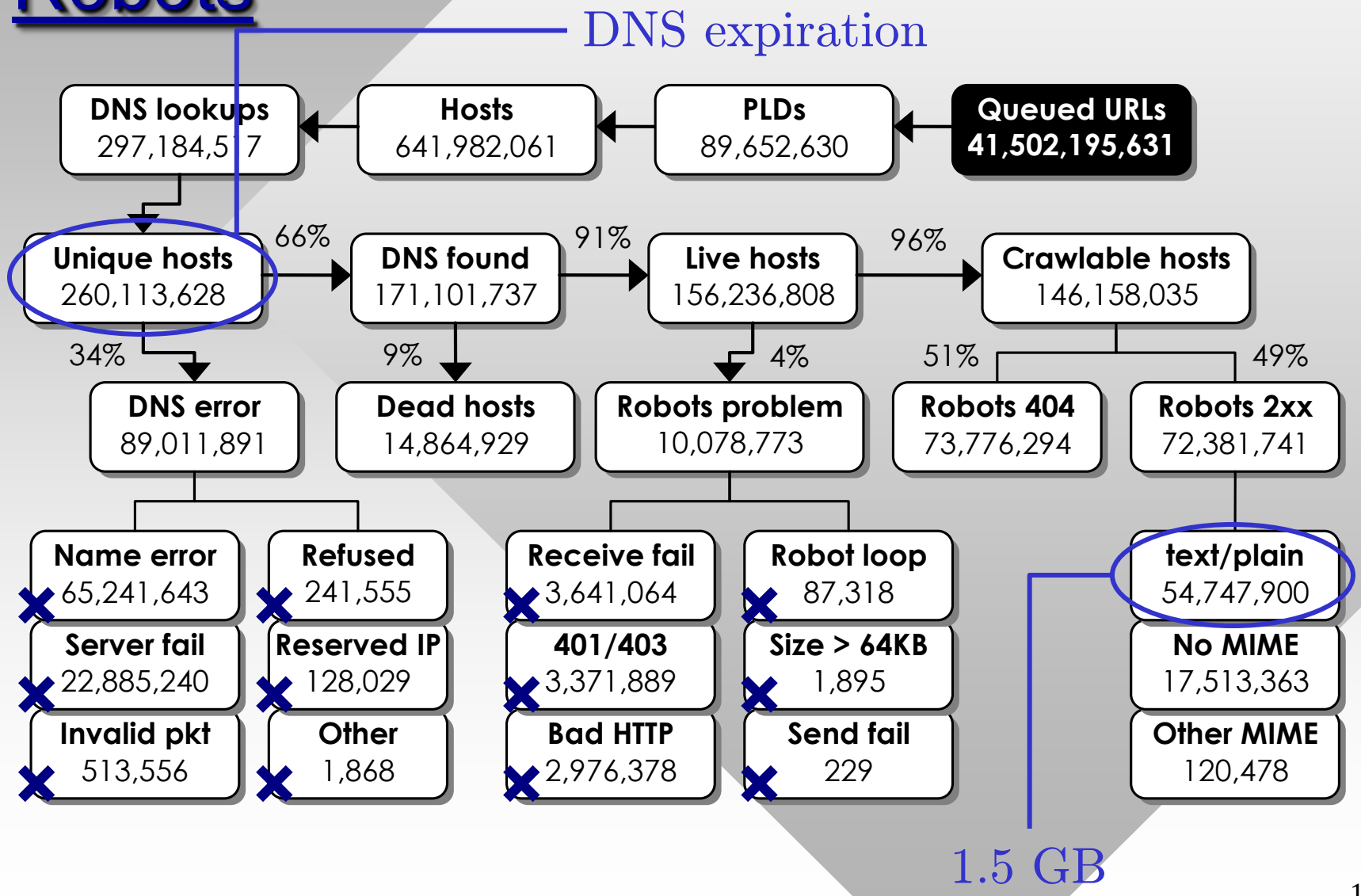
# A Few Notes (2)

- URL Processing
  - Processed a-href, frame-src and meta-refresh tags. Did not follow img tags
  - Checked URLs for correctness and syntax
  - Used a black list of non-HTML extensions, resulted in $0.37\%$ saving in bandwidth (note for future crawlers)
- Web graph
  - Constructed a web graph with $3$ TB web graph with $310$B edges and $41$B nodes
  - Average crawl depth $12$ (compare to $1.8$ of ClueWeb09)
  - 60% of downloaded pages were dynamic (i.e. contains “?”)

# Agenda

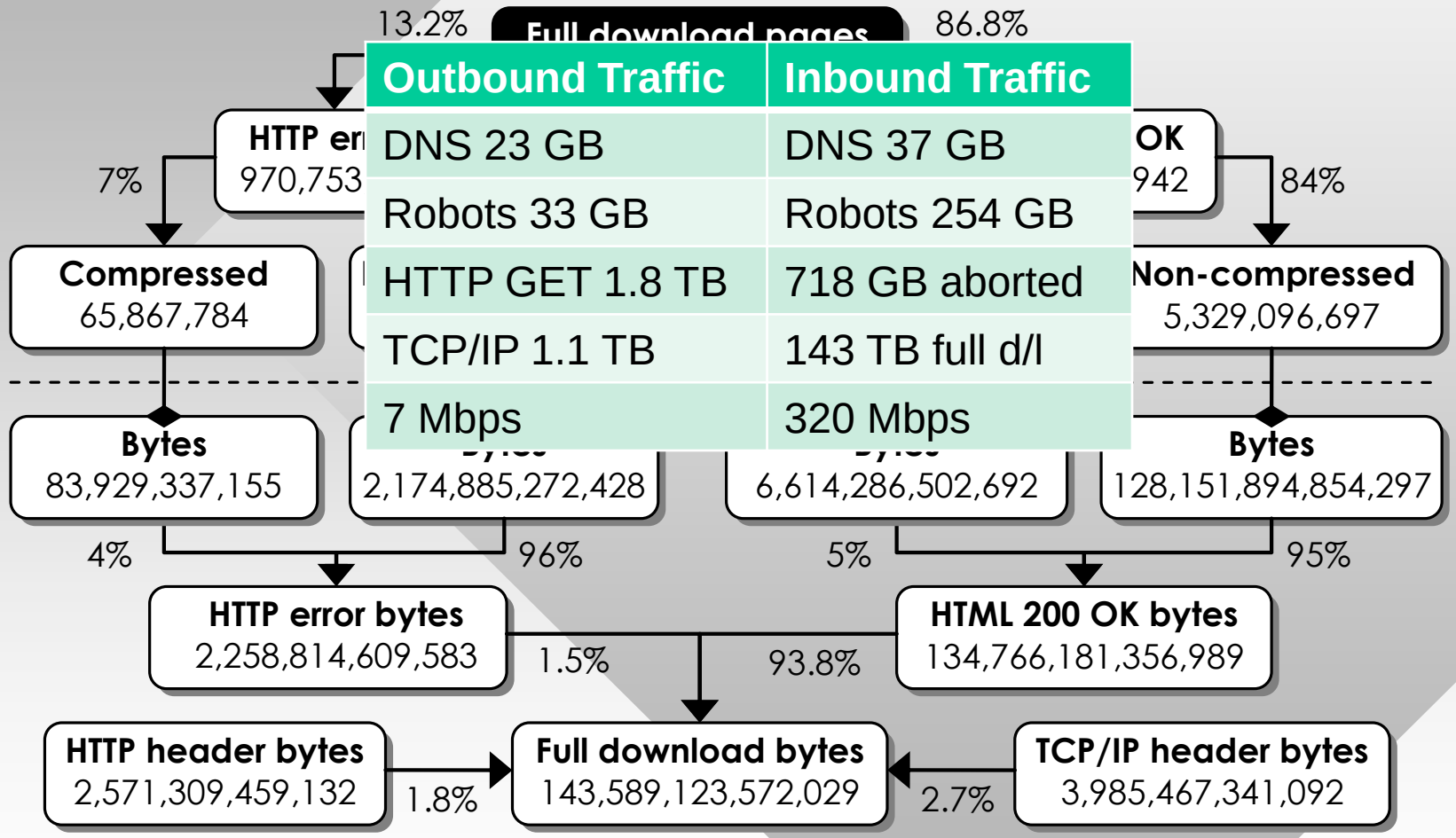
- Introduction
- Background
- Page-Level Analysis
- Server-Level Analysis
  - DNS and Robots
  - Bandwidth
- Internet Coverage
- Extrapolation
- Conclusion

# Server-Level Analysis – DNS and Robots



# Server-Level Analysis - Bandwidth

| Outbound Traffic | Inbound Traffic |
|---|---|
| DNS 23 GB | DNS 37 GB |
| Robots 33 GB | Robots 254 GB |
| HTTP GET 1.8 TB | 718 GB aborted |
| TCP/IP 1.1 TB | 143 TB full d/l |
| 7 Mbps | 320 Mbps |



# Agenda

- Introduction
- Background
- Page-Level Analysis
- Server-Level Analysis
- Internet Coverage
- Extrapolation
- Conclusion

# Internet Coverage

- Can use different measures
  - Collection of crawled 200 OK pages
  - Constructed web graph size
- Not much available information in standardized fashion
  - Mercator uses img tags, while UbiCrawler removes frontiers
  - WebBase considers robots.txt as crawled page

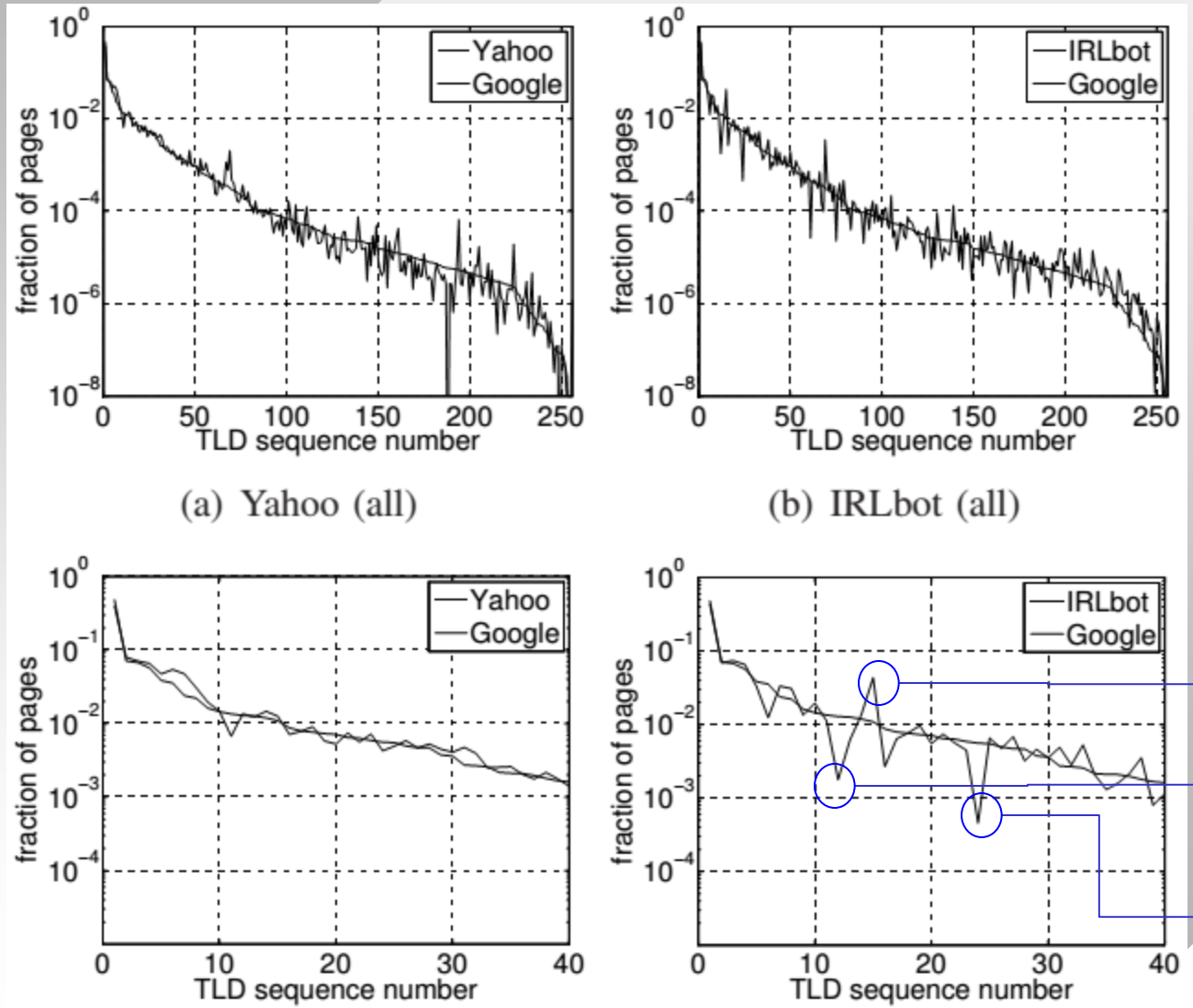
| Dataset | Crawled (HTML 200 OK) | | | | Web graph | | Host graph | | PLD graph | |
|---|---|---|---|---|---|---|---|---|---|---|
| | pages | hosts | PLDs | TLDs | nodes | edges | nodes | edges | nodes | edges |
| AltaVista [9] | – | – | – | – | 271M | 2.1B | – | – | – | – |
| Polybot [36] | 121M | 5M | – | – | – | – | – | – | – | – |
| Google [6] | – | – | – | – | 1.3B | 19.5B | 12.8M | 395M | – | – |
| Mercator [10] | 429M | ∼ 10M | – | – | – | 18.3B | – | – | – | – |
| WebFountain [20] | 1B | – | – | – | 4.75B | 37B | 19.7M | 1.1B | – | – |
| WebBase [16] | 98M | 51K | – | – | – | 4.2B | – | – | – | – |
| ClueWeb09 [19] | 1B | – | – | – | 4.8B | 7.9B | – | – | – | – |
| IRLbot | 6.3B | 117M | 33M | 256 | 41B | 310B | 641M | 6.8B | 89M | 1.8B |
| UbiCrawler .uk [7] | 105M | 114K | – | 1 | 105M | 3.7B | 114K | – | – | – |
| IRLbot .uk | 197M | 2.8M | 1.2M | 1 | 1.3B | 9.5B | 5M | 54M | 1.5M | 18M |
| TeaPot .cn [41] | 837M | 16.9M | 790K | 1 | 837M | 43B | 16.9M | – | 790K | – |
| IRLbot .cn | 209M | 3.3M | 539K | 1 | 1.1B | 11.9B | 8.4M | 103M | 711K | 19.7M |

# Internet Coverage – TLD Level

- A novel method of comparing crawls
  - Reveals crawler budget on different parts of the Internet
- Use site queries (i.e., "site:domain") to obtain Google and Yahoo's (now part of Bing) index size
  - In 1/2008, they contained 30B and 37B pages, respectively

| TLD | Google | Yahoo | IRLbot | WebBase | ClueWeb |
|---|---|---|---|---|---|
| .com | 46.7% | 38.3% | 43.3% | 31.2% | 54.8% |
| .net | 6.9% | 7.7% | 6.9% | 2.2% | 6.7% |
| .de | 6.6% | 6.8% | 7.4% | 3.8% | 3.8% |
| .org | 5.5% | 6.3% | 6.6% | 17.8% | 6.6% |
| .cn | 3.7% | 4.6% | 3.3% | 0.2% | 5.6% |
| .jp | 3.4% | 5.2% | 1.2% | 1.7% | 3.2% |
| .ru | 2.3% | 4.6% | 3.3% | 0.6% | 0.1% |
| .uk | 2.2% | 3.0% | 3.1% | 4.9% | 1.7% |
| .pl | 1.6% | 1.9% | 1.3% | 0.2% | 0.3% |
| .nl | 1.4% | 1.4% | 2.0% | 0.5% | 0.1% |
| TLDs | 255 | 256 | 256 | 174 | 254 |

# TLD Coverage – Google Order



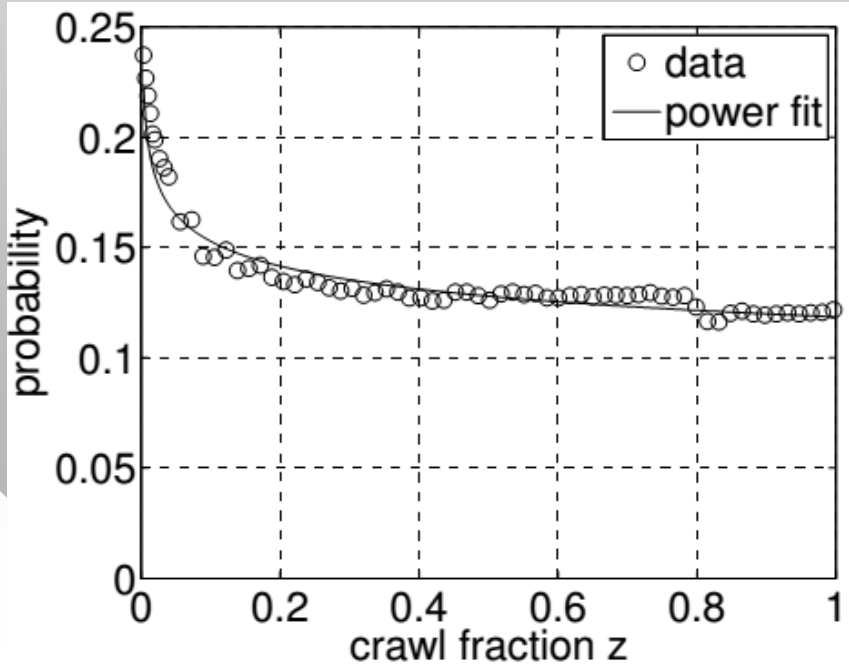
# Agenda

- Introduction
- Background
- Page-Level Analysis
- Server-Level Analysis
- Internet Coverage
- Extrapolation
- Conclusion

# Extrapolation

- Assume that the crawl is stochastic process $\{(X_t, Y_t)\}$ on the Internet, a web graph $G(V,E)$, where the process terminates at $t = N \leq |E|$ edges
- Define $p(t)$ as the probability that URL $Y_t$ has not been seen before
- Objective: In a larger crawl, can we estimate number of
  - Unique URLs $L_N$
  - Crawled pages $C_N = L_N \langle l \rangle$

.# of links/page



# Extrapolation (2)

- Assume that the reference crawl (IRLbot) has $\mathcal{K}$ links, $\mathcal{U}$ unique links. The unknown crawl (e.g., Google) has $N$ links ($r=N/\mathcal{K}$). What is $L_N$ and $C_N$ ?
- Also assume $z=t/\mathcal{K}$ and a new function $\tilde{p}(z) = p(z\mathcal{K})$ . Thus, the unknown crawler has:

$$E[L_N] = \mathcal{K} \int_0^r \tilde{p}(z)dz = \mathcal{U} + \mathcal{K} \int_1^r \tilde{p}(z)dz$$

- With Pareto fit (i.e., $\tilde{p}(z) = \beta z^{-\alpha}$ ), we get:

$$E[L_N] \approx \mathcal{U} + \frac{\mathcal{K}\beta(r^{1-\alpha} - 1)}{1-\alpha}$$

# Extrapolation - Results

| Crawl | Ratio $r$ | Crawled Links $N$ | Crawled Pages $C_N$ |
|---|---|---|---|
| IRLbot 2007 | 1 | 394B | 6.3B |
| Google 2008 ($E[L_N] = 1\text{T}$) | 40 | 12T | 256B |
| Google 2012 ($E[L_N] = 30\text{T}$) | 1,981 | 592T | 12T |

Using 20B pages/day (@41 Gbps), takes 50 months of crawling

- How about Hots/PLD level graphs in Google 2012?
  - With $r$=1981, Google has 5.2B unique hosts (IRLbot has 641M), and 90.6M unique PLDs (IRLbot has 89M)

# Conclusion

- Presented IRLbot implementation and experiment in detail
  - Discussed the impact of various design choices
  - Provided guidelines for future crawlers
  - Exposed weird/effective spamming techniques
- Developed new methods for capturing crawl coverage
- Outlined a simple extrapolation mechanism to infer proprietary and undocumented crawls
  - A simple model for crawl growth rate

# Thank you!
# Questions?