FeedEx: Collaborative Exchange of News Feeds

Seung Jun and Mustaque Ahamad
Georgia Tech Information Security Center
Georgia Institute of Technology

Motivation

- RSS/Atom feeds have become increasingly popular
  - Published by most traditional media and blogs
- Scalability of feed servers
  - Frequent pull requests create high load
  - Infrequent requests increase latency and may lead to missed items
- Our Approach
  - Use resources at peer nodes to deliver feed items
  - Scalable growth in resources with service demand
- Challenges
  - Peers may not fully cooperate and execute the agreed protocols

FeedEx Overview

- Feeds have different update and usage patterns.
  - A new hybrid transport mechanism
  - Pull from servers
  - Push among peer nodes
- Peers in FeedEx
  - Form a distribution mesh,
  - Fetch feeds from web servers occasionally, and
  - Exchange new entries among each other
  - Peer incentives for exchanging entries

RSS/Atom Primer

- Feed format

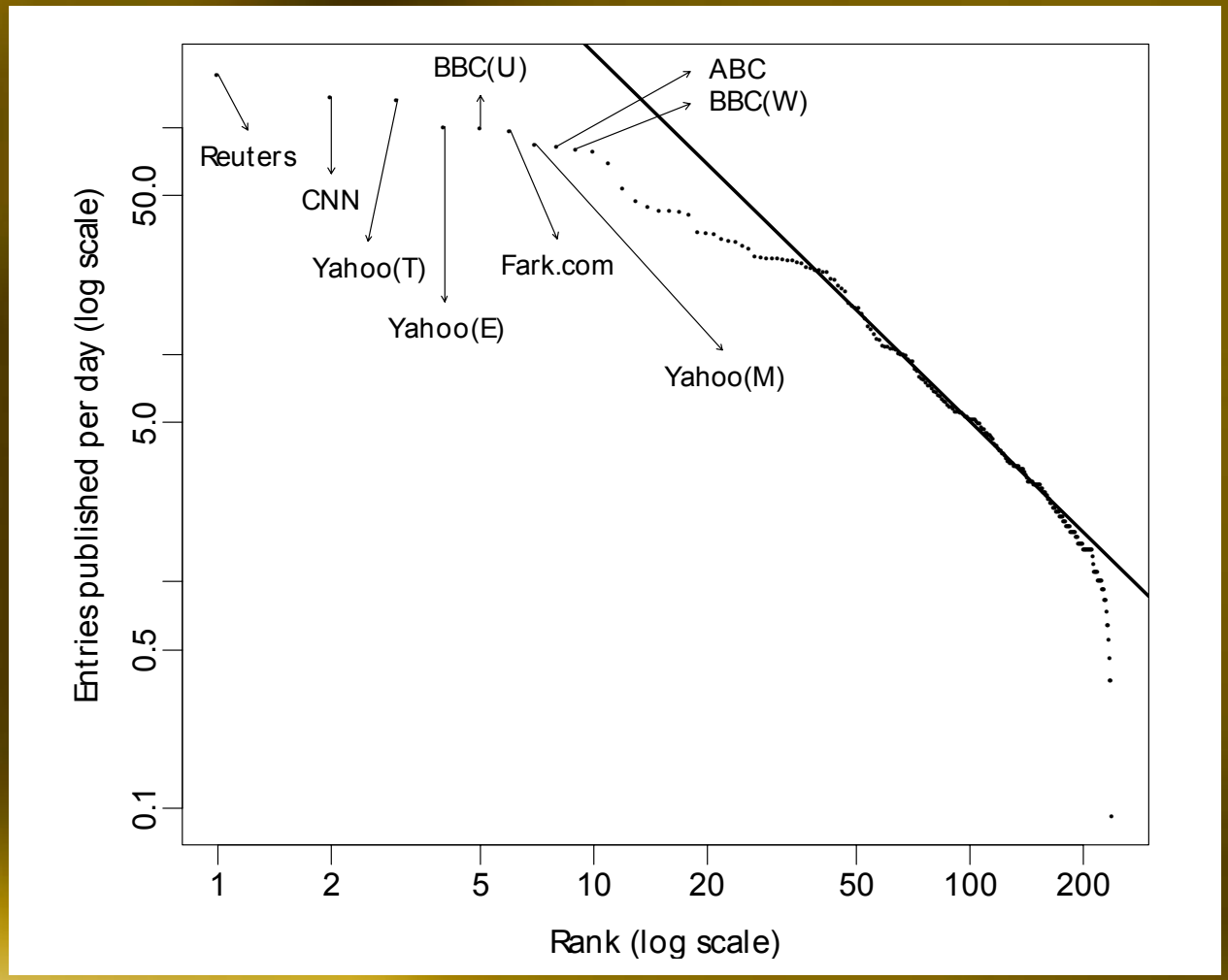
```
<feed>
   <title>NYT Technology</title>
   <!-- other elements -->
   <entry>
     <title>Basics: Going Wireless on ...</title>
     <link>http://www.nytimes.com/2006/05/18/...</link>
     <summary>Wi-Fi has revolutionized the...</summary>
     <!-- other elements -->
   </entry>

   <!-- more entries -->
</feed>
```

- Current way of reading feeds
  - Stand-alone applications (e.g., Mozilla Thunderbird)
  - Web-based service (e.g., Bloglines and My Yahoo!)

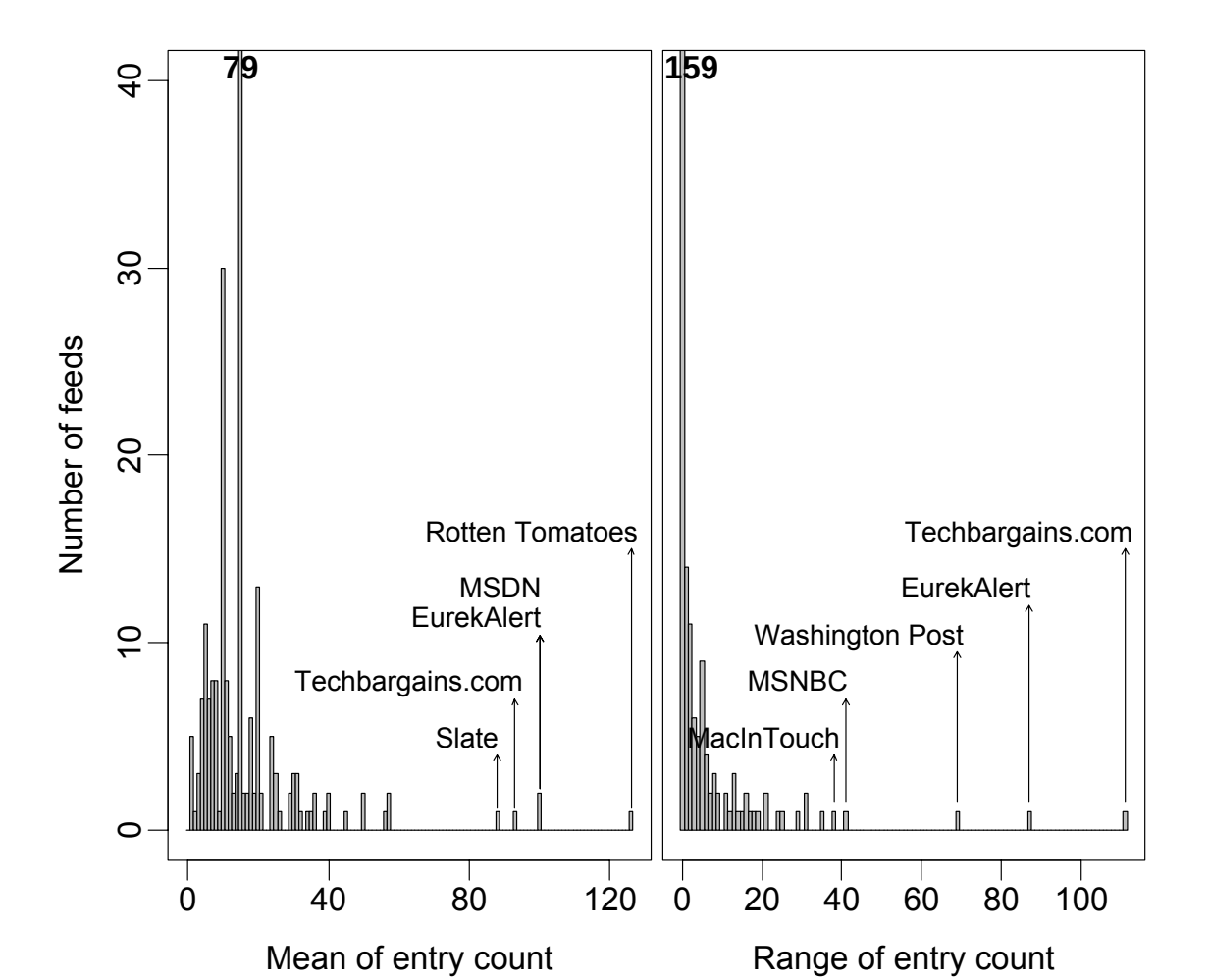
Analysis of Feed Publishing

- Purpose
  - Interesting by itself and helpful in designing FeedEx
- Methodology
  - 245 popular feeds monitored for 10 days
  - Feeds fetched every 2 minutes

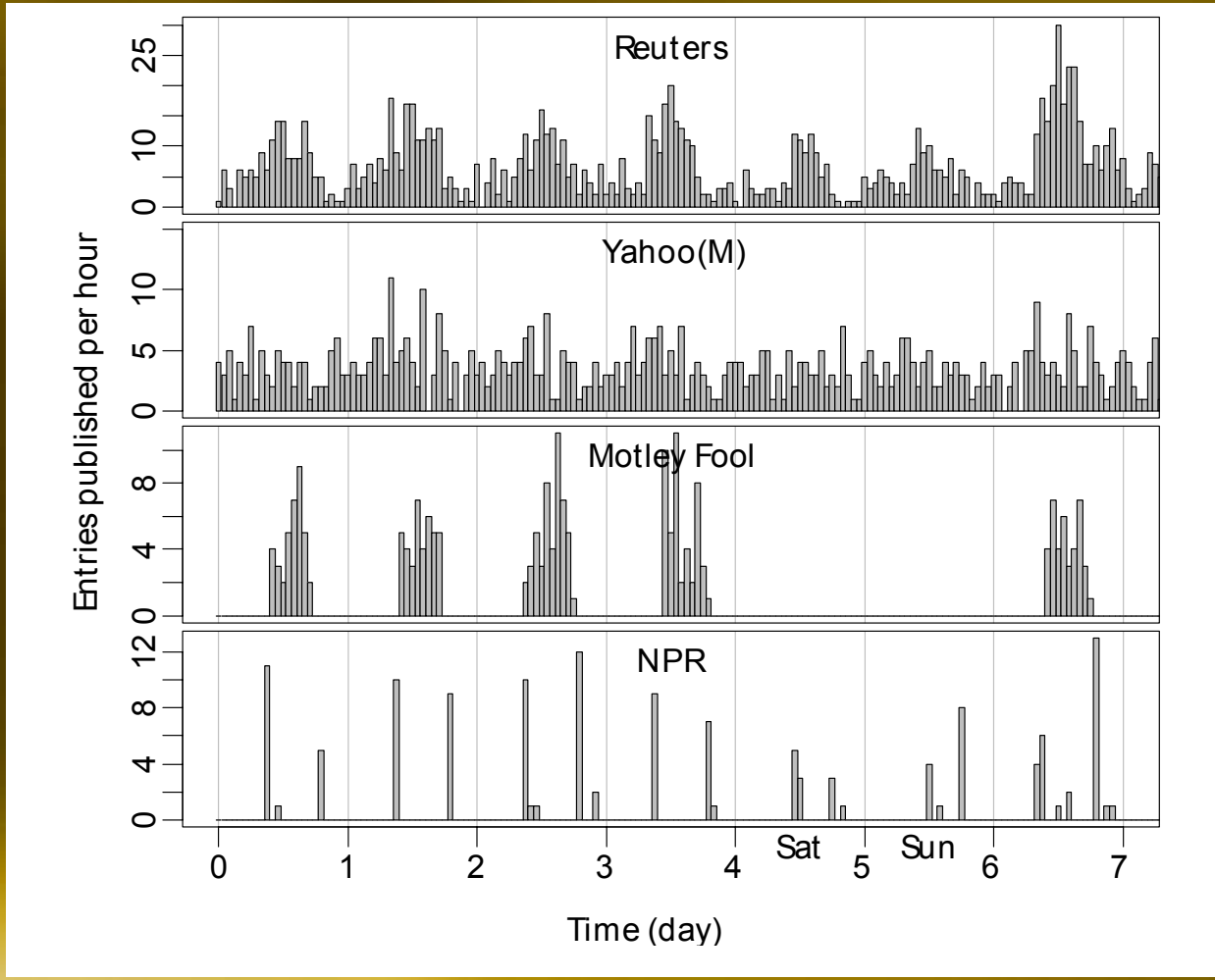
Publishing Rate by Rank



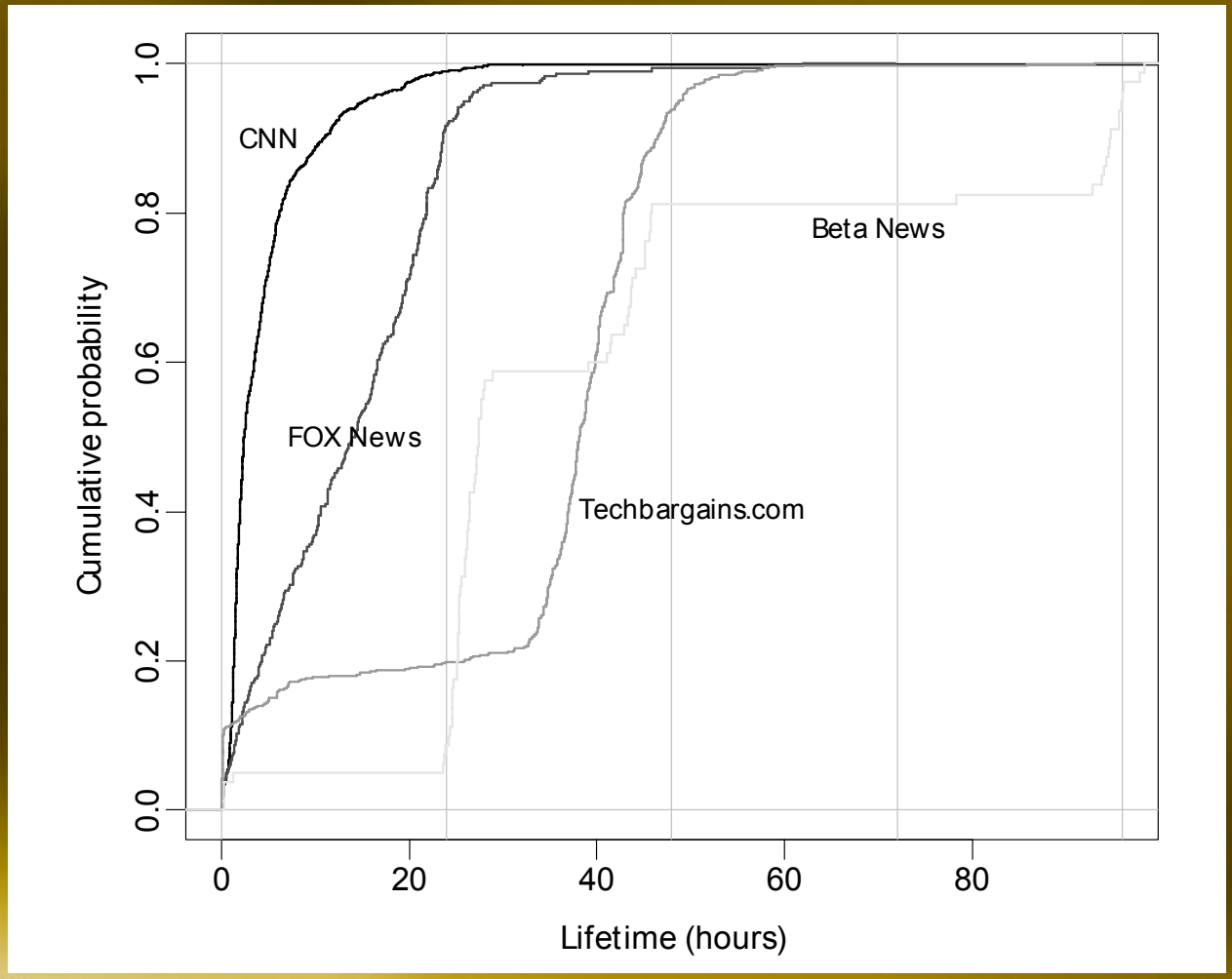
Entry Count


79
159
Number of feeds
0
10
20
30
40
Rotten Tomatoes
MSDN
EurekAlert
Techbargains.com
Slate
Techbargains.com
EurekAlert
Washington Post
MSNBC
MacInTouch
0
40
80
120
0
20
40
60
80
100
Mean of entry count
Range of entry count

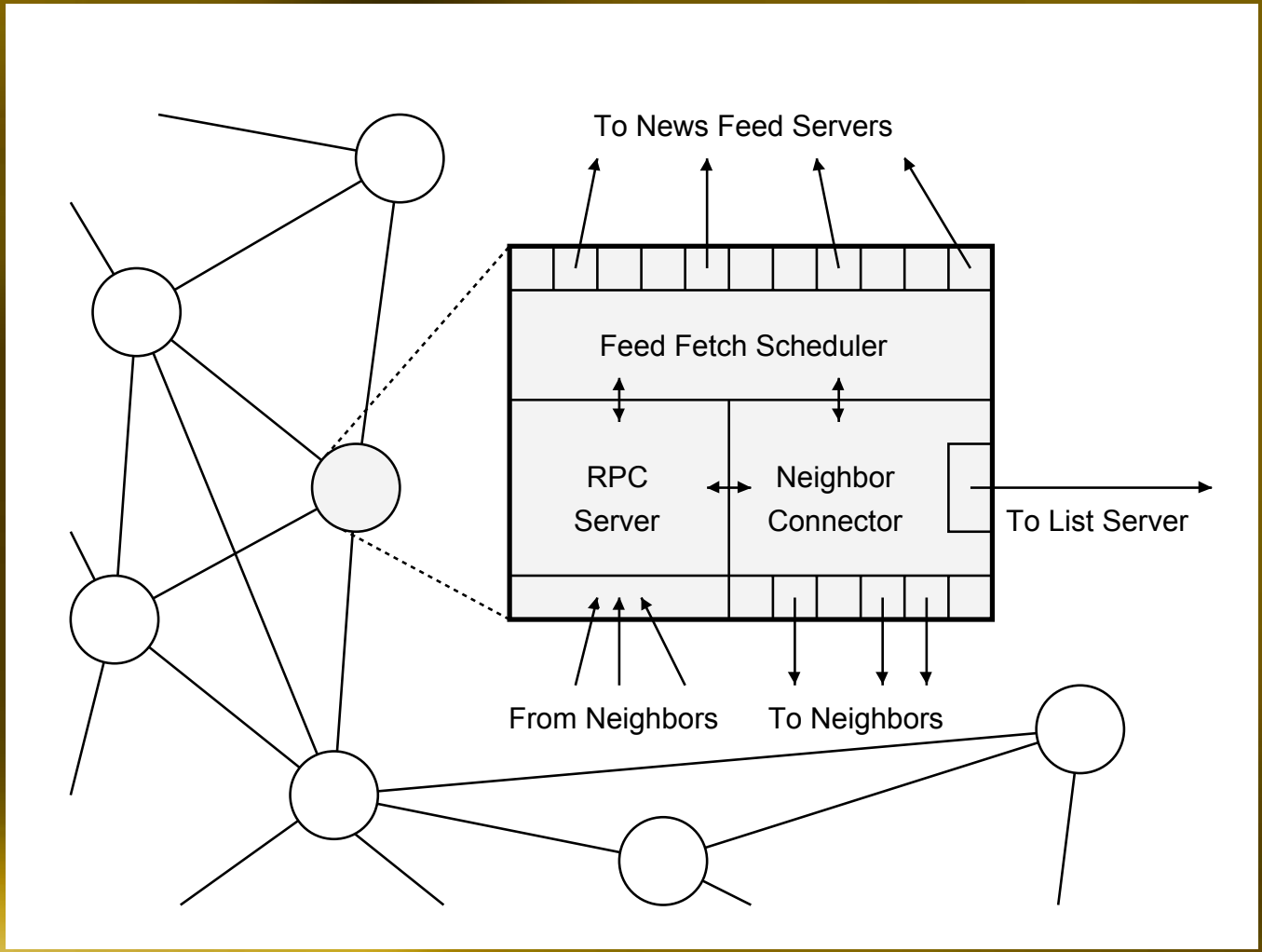
Publishing Rate by Time



Entry Lifetime



Architecture of FeedEx



Bootstrapping

- Obtain a list of peers
  - Dedicated list server (Gnutella and BitTorrent)
  - Embedding (Pseudoserving [Kong and Ghosal 1999] and CoopNet [Padmanabhan and Sripanidkulchai 2002])
  - Local cache
- Connect to peers
  1. Establish connection
  2. Exchange subscription sets: {(url,hop),...}

Neighbor Selection

- Metrics for good neighbors
  - Subscription set match

$$u(Q) = \sum_{i \in (S_P \cap S'_Q)} w_i d^{-h_i}$$

  - Topological proximity
  - Duration of relationship

Adaptive Fetching from Servers

- Coordinated fetching by peers
  - High coordination overhead
  - Lots of nodes with high churn rate
- Solution: Adaptive fetching
  - Freshness rate f: Fraction of new entries in a fetched document
  - Set a target freshness rate f_t
  - Fetching interval is doubled or halved, bounded by T_{min} and T_{max}

Entry Exchange Among Peers

- New entries obtained
  - By fetching from web servers
  - From neighbors
- Entry bundle
  - A set of new entries
  - Document identifier (did): Assigned by SHA-1 digest
  - Flooded to matching neighbors
- Two-phase flooding
  - `check_did(did)` call: 344 bytes including HTTP request header
  - `put_entries(bundle)`

Incentive Mechanism

- Pairwise fairness is simple and effective
  - Uses local information only
  - Easy to implement and enforce the mechanism
- Contribution metric $c_{j,i}$:

$$c_{j,i} += w_f \cdot hf$$

- Deficit of contribution $d_{i,j}$:

$$d_{i,j} = c_{i,j} - c_{j,i}$$

- Node i ensures $d_{i,j} < D$ for every neighbor j and a parameter D.

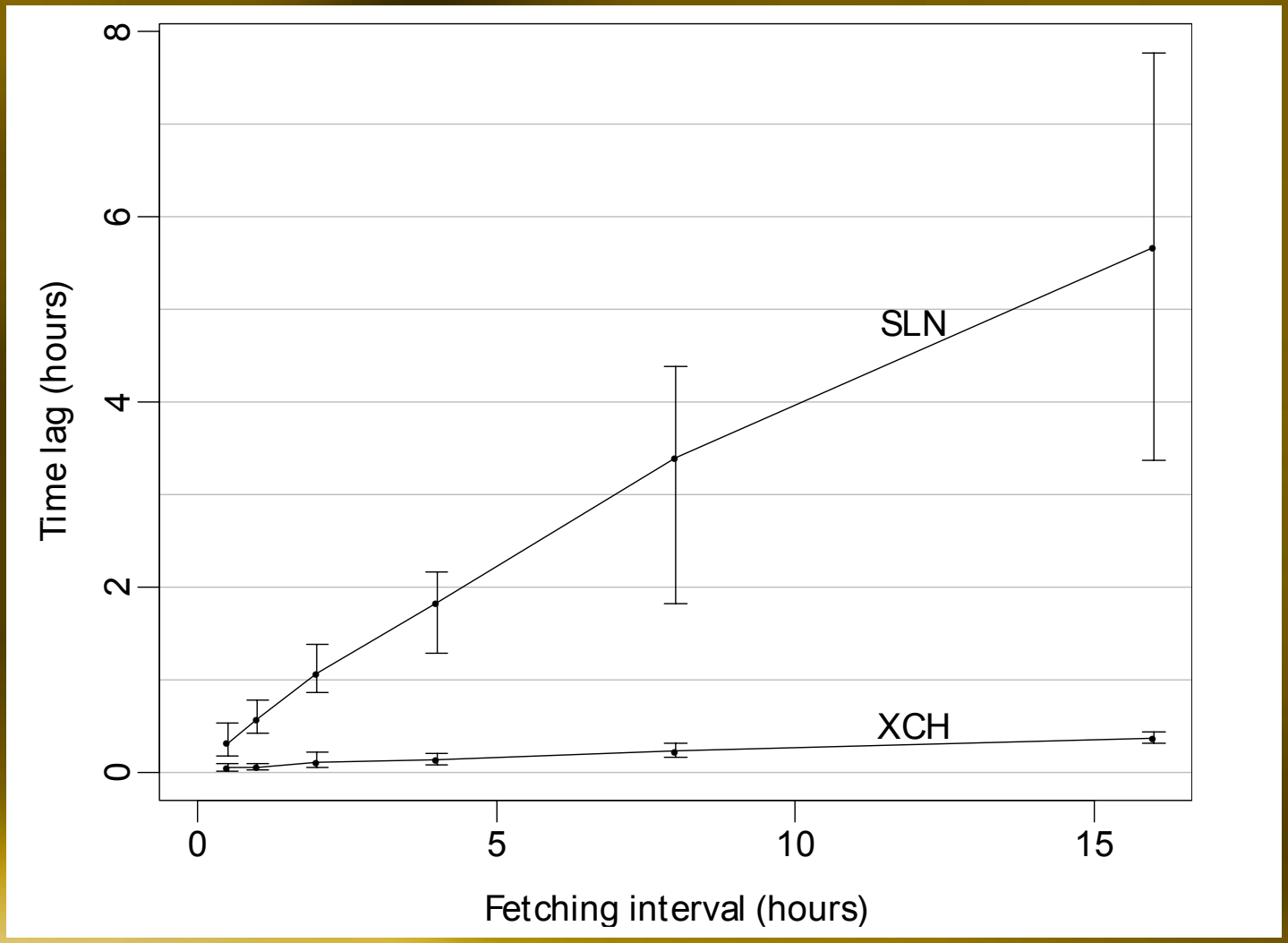
Prototype Implementation

- Python: `python.org`
- XML-RPC: `xmlrpc.com/spec`
- Twisted: `twistedmatrix.com`
- SQLite: `sqlite.org`
- Universal Feed Parser: `feedparser.org`

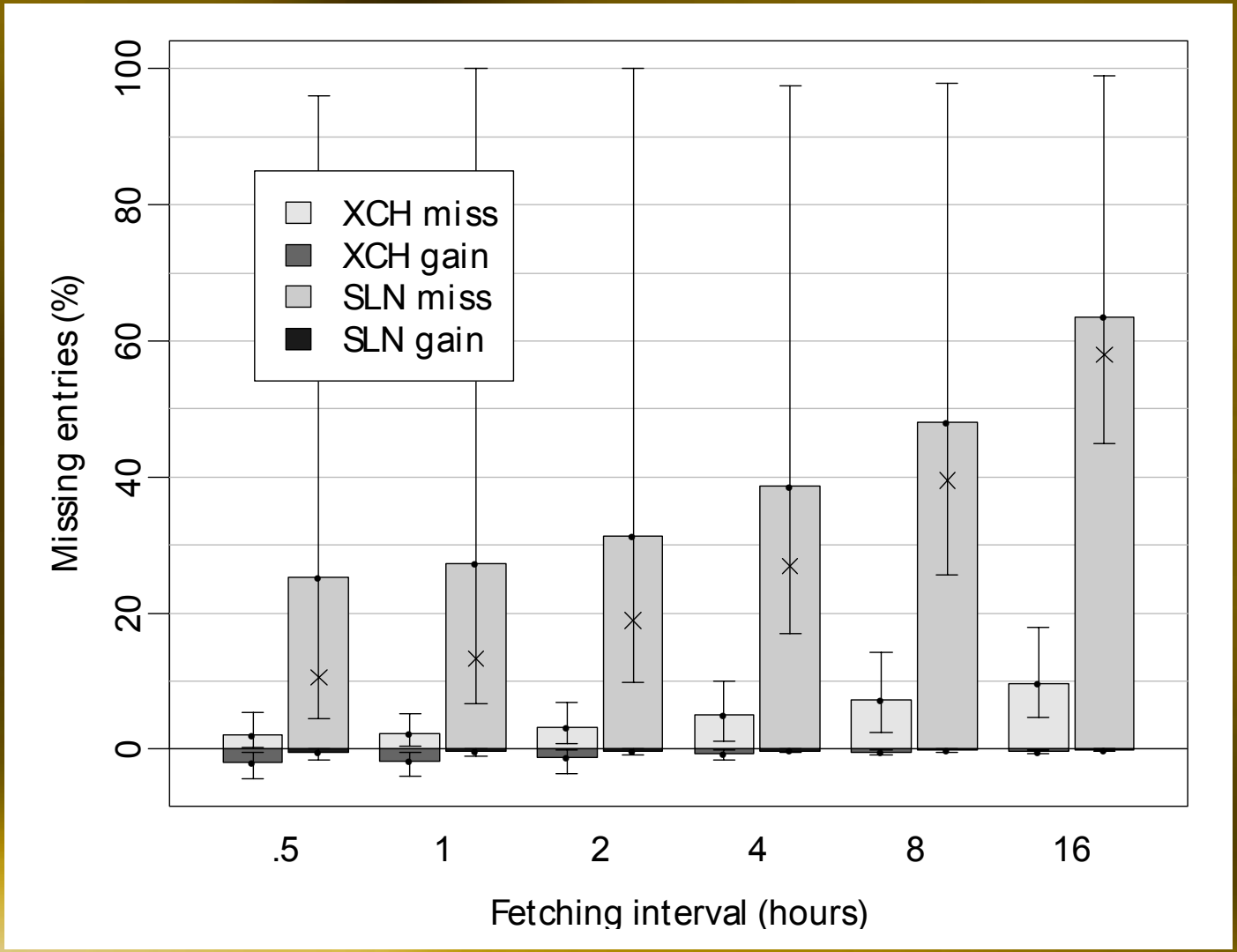
Experimental Setup

- Two modes
  - Stand-alone applications: `sln`
  - FeedEx: `xch`
- Metrics
  - Time lag
  - Missing entries
  - Communication cost
- Experiments
  - Use 189 PlanetLab nodes
  - Run 22 hours on a weekday
  - Primary factor: 6 fetching intervals
  - Let each node subscribe 20 out of 70 feeds

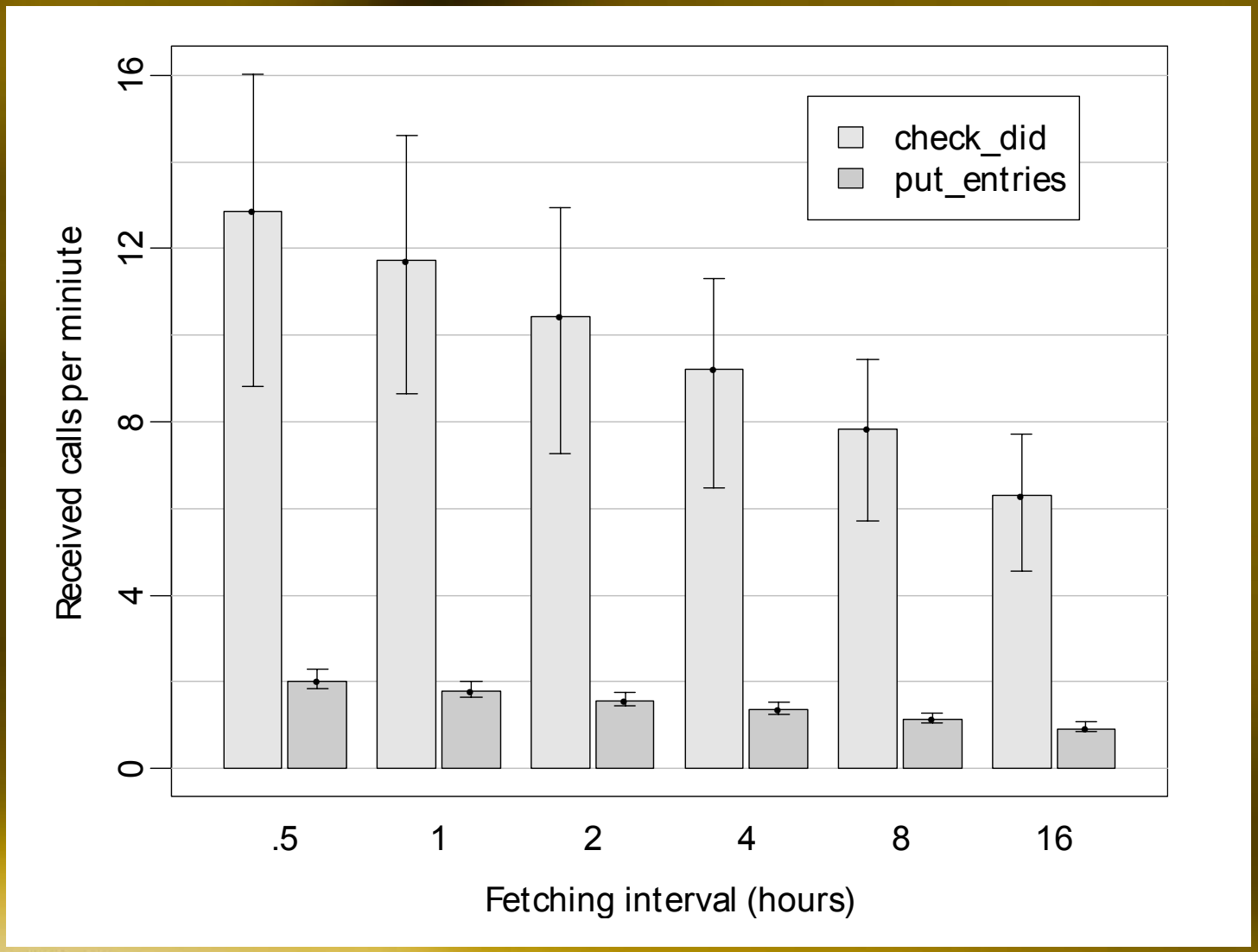
Results: Time Lag



Results: Missing Entries



Results: Communication Cost



Advantages

- Server scalability
- Archivability
- Controllability
- Filtering and recommendation
- Privacy

Related Work

- News feed delivery
  - Corona (Cornell)
  - FeedTree (Rice)
- Web caching and CDN [Freedman et al. 2004, Wang et al. 2004]
- Gossip-based protocols [Birman et al. 1999, Ganesh et al. 2003, Eugster et al. 2003]

Conclusions

- A new transport mechanism for news feeds
  - Pull by and exchange among peers
- FeedEx encourages cooperation by enforcing pair-wise fairness, while achieving
  - Reduced feed server load
  - Low latency
  - High coverage
  - Low communication overhead