Approximate Object Location and Spam Filtering on Peer-to-Peer Systems

Feng Zhou, Li Zhuang, Ben Y. Zhao,
Ling Huang, Anthony D. Joseph
and John D. Kubiatowicz

University of California, Berkeley

The Problem of Spam

- Spam
  - Unsolicited, automated emails
  - Radicati Group: $20B cost in 2003, $198B in 2007
- Proposed solutions
  - Economic model for spam prevention
    - Attach cost to mass email distribution
    - Weakness: needs wide-spread deployment, prevent but not filter
  - Bayesian network / machine learning (independent)
    - “Train” mailer with spam, rely on recognizing words / patterns
    - Weakness: key words can be masked (images, invis. characters)
  - Collaborative filtering
    - Store / query for spam signatures on central repository
    - Other users query signatures to filter out incoming spam
    - Weakness: central repository limited in bandwidth, computation

Our Contribution

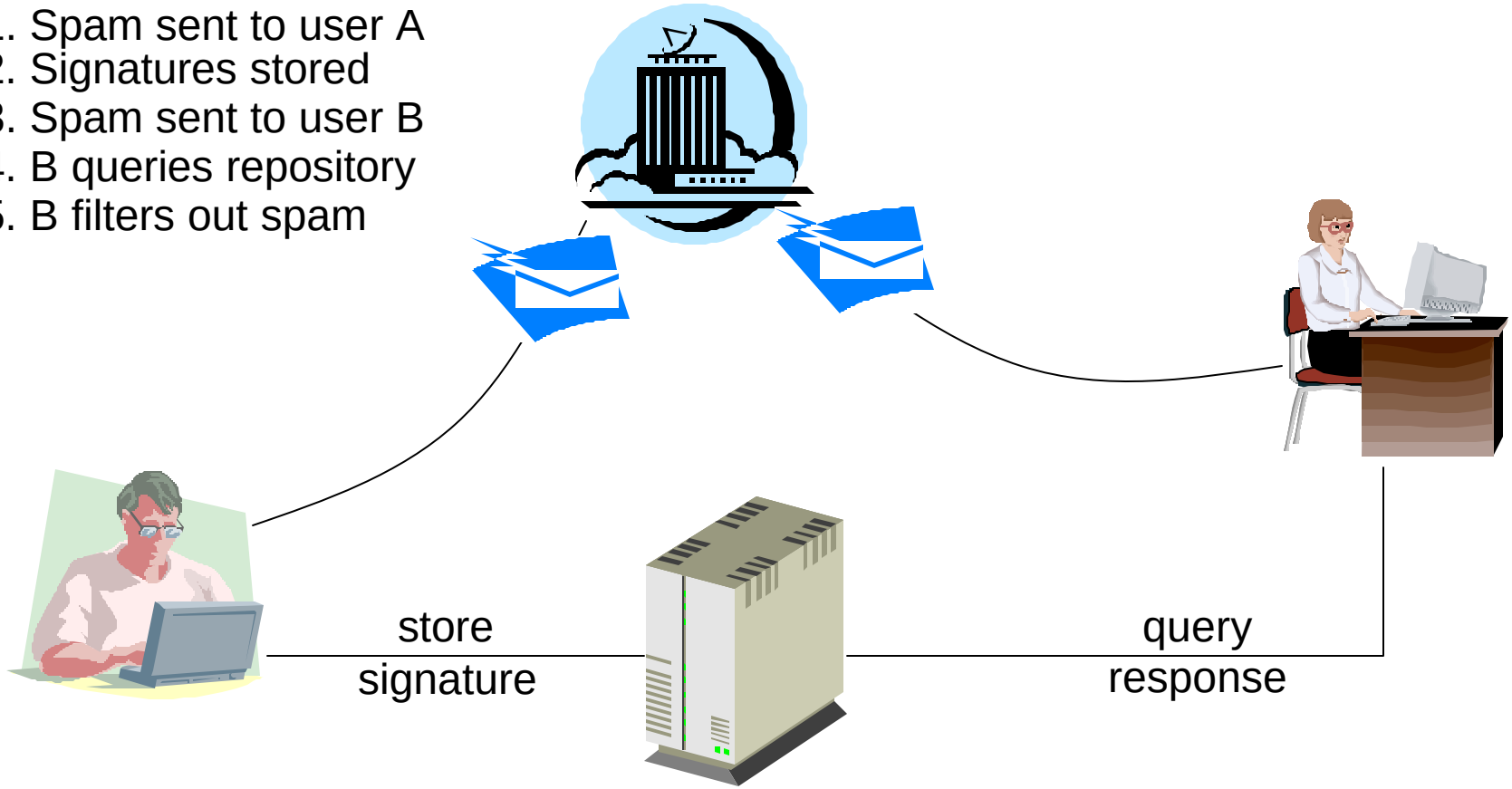
- Can signatures effectively detect modified spam?
  - Goals:
    - Minimize false positives (marking good email as spam)
    - Recognize modified/customized spam as same as original
  - Present signature scheme based on approx. fingerprints
  - Evaluate against random text and real email messages
- Can we build a scalable, resilient signature repository
  - Leverage structured peer-to-peer networks
  - Constrain query latency and limit bandwidth usage
- Orthogonal issues we do not address:
  - Preprocessing emails to extract content
  - Interpreting collective votes via reputation systems

Outline

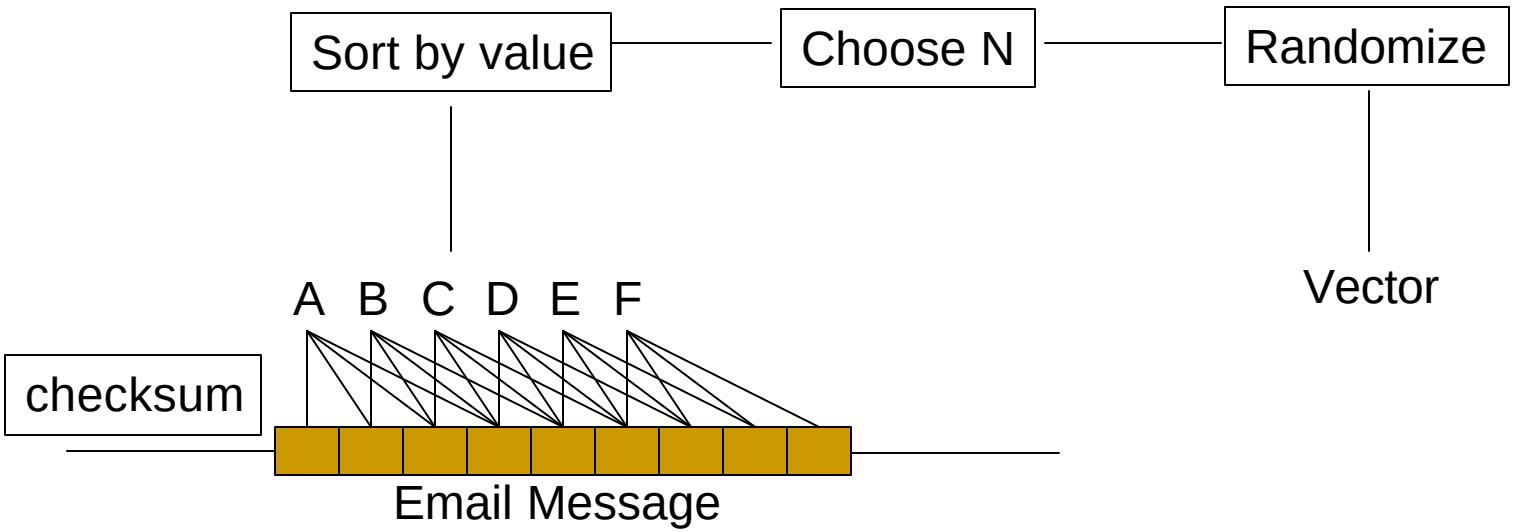
- Introduction
- An Approximate Signature Scheme
  - Evaluation using random text and real emails
- Approximate object location
  - Similarity search on P2P systems
  - Constraining latency and bandwidth usage
- Conclusion

Collaborative Spam Filtering

1. Spam sent to user A
2. Signatures stored
3. Spam sent to user B
4. B queries repository
5. B filters out spam

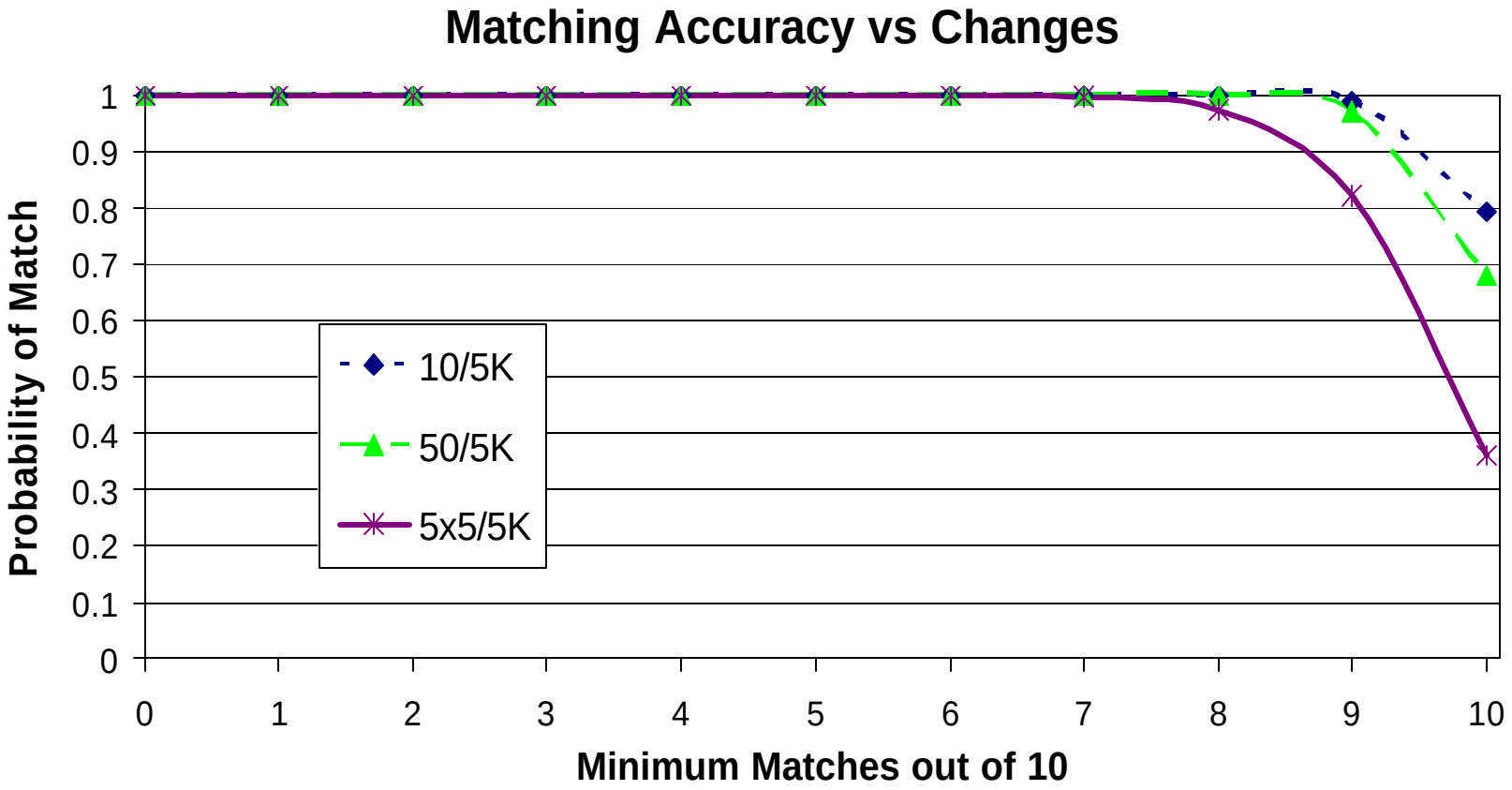


An Approximate Signature Scheme



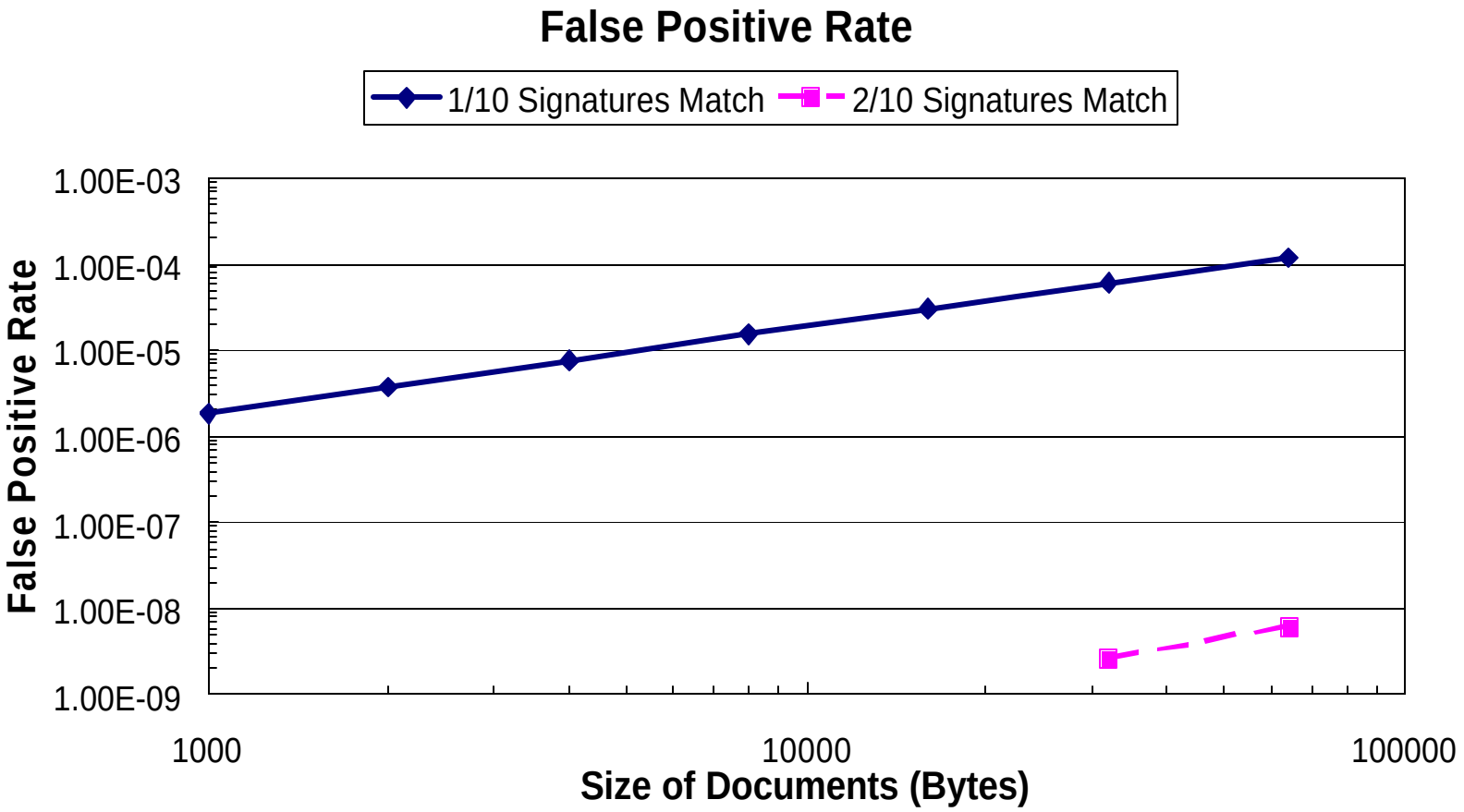
- How to match documents with very similar content
- Calculate checksums of all substrings of length L
- Select deterministic set of N checksums
- *A matches B* ***iff*** $|sig(A) \cap sig(B)| > Threshold$
- Computation tput: 13MByte/s on P-III 1Ghz

Accuracy of Signature Vectors



- 10000 random text documents, size = 5KB, calculate 10 signatures
- Compare signatures of before and after modifications
- Analytical results match experimental results

Eliminating False positives



- Compare pair-wise signatures between 10000 random docs
- None matched 3 of 10 signatures (100,000,000 pairs)

Evaluation on Real Messages

- 29631 Spam Emails from www.spamarchive.org
  - Processed visually by project members
  - 14925 (unique), 86% of spam = 5K
- Robustness to modification test
  - Most popular 39 msgs have 3440 modified copies
  - Examine recognition between copies and originals

THRES	Detected	Failed	%
3/10	**3356**	**84**	**97.56**
4/10	3172	268	92.21
5/10	2967	473	86.25

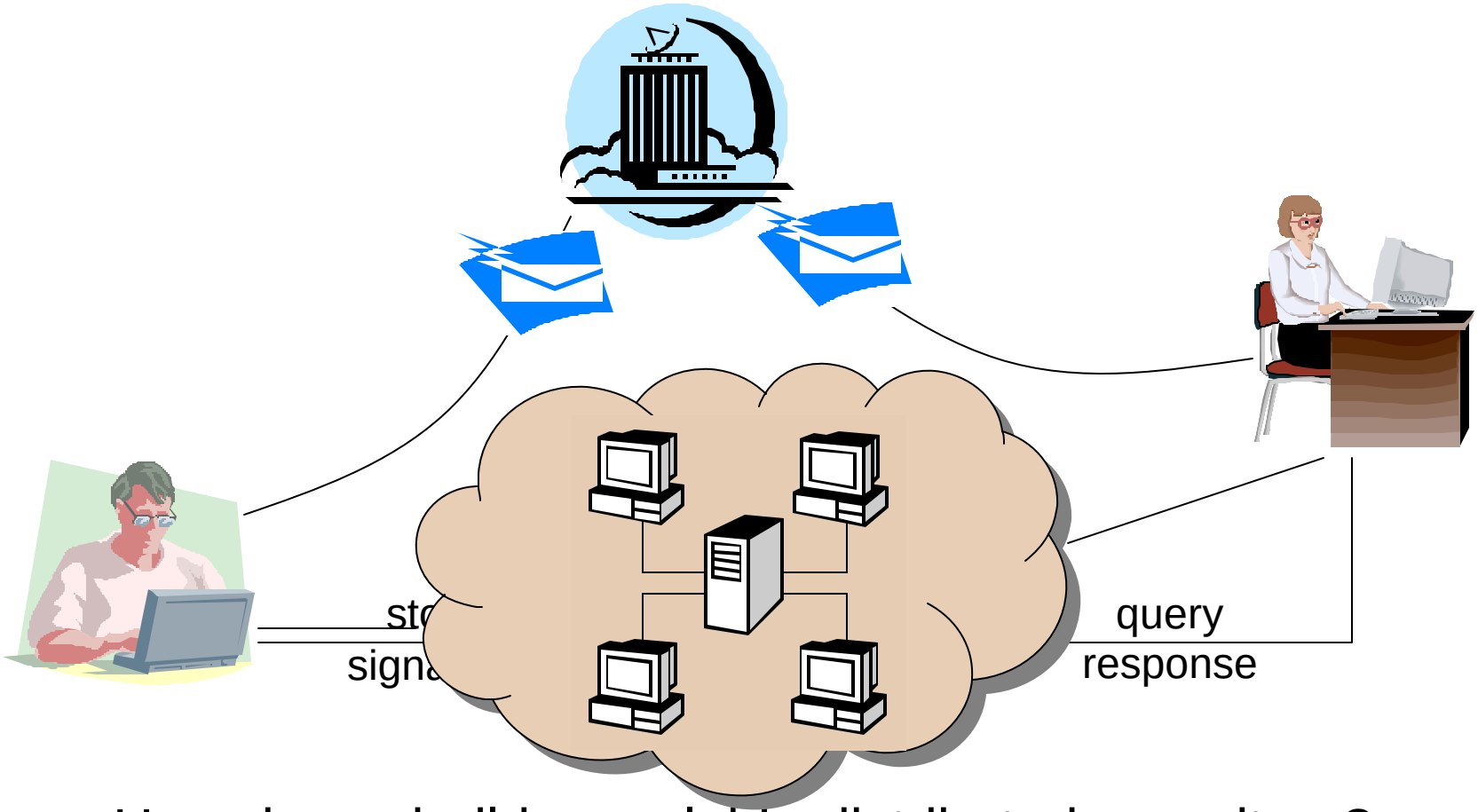
False Positive Test

- Non-spam emails
  - 9589 messages: 50% newsgroup posts + 50% personal emails
  - Compare against 14925 unique spam messages

THRES	# of pairs	Probability
1/10	270	1.89e-6
2/10	4	2.79e-8
3/10	**0**	**0**

- Sweet spot, using threshold of 3/10 signatures
  - Recognition rate > 97.5%
  - False positive rate < 1 in 140 million pairs

A Distributed Signature Repository?

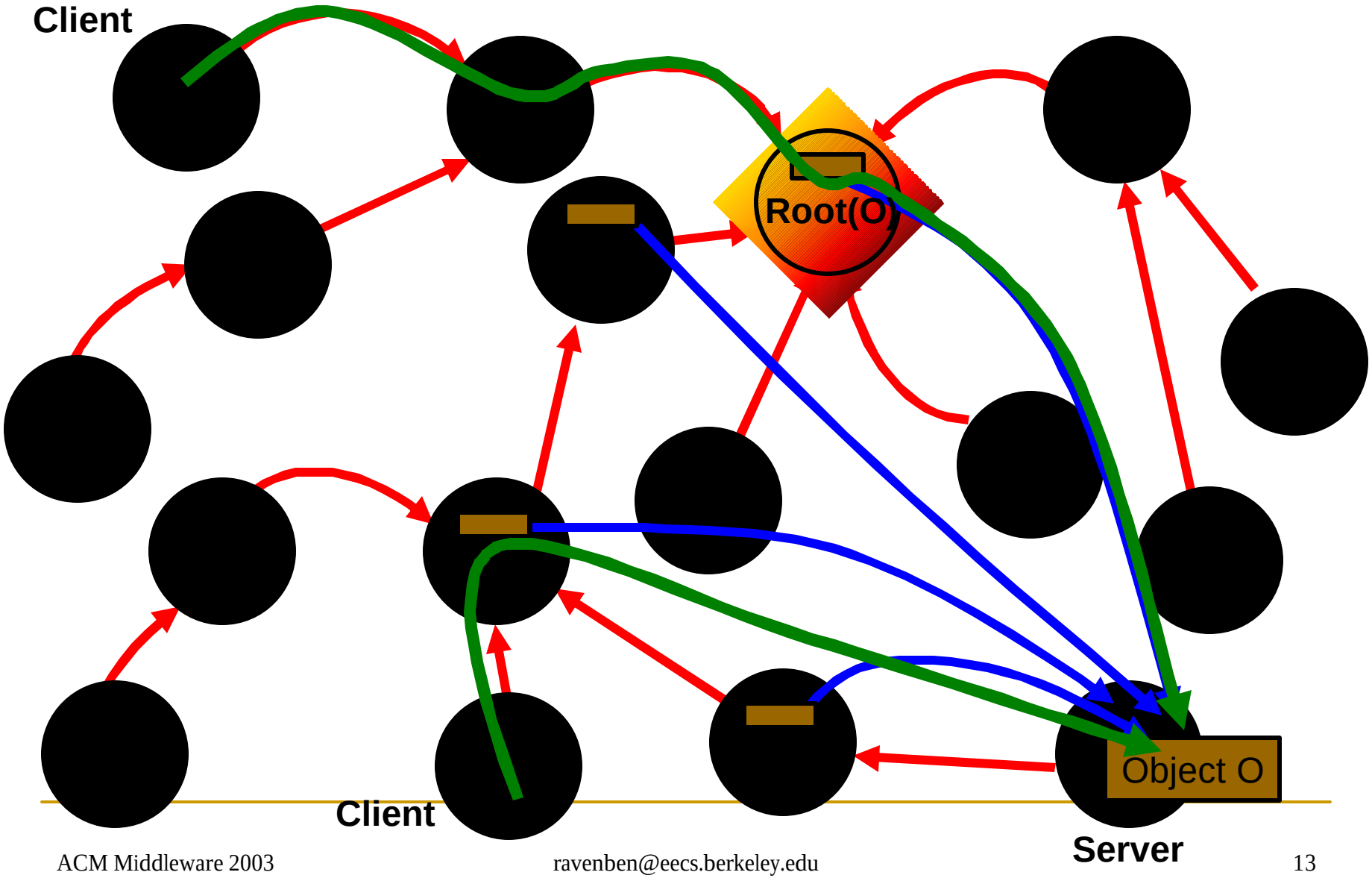


- How do we build a scalable distributed repository?
- How do we limit bandwidth consumption and latency?

Structured Peer-to-Peer Overlays

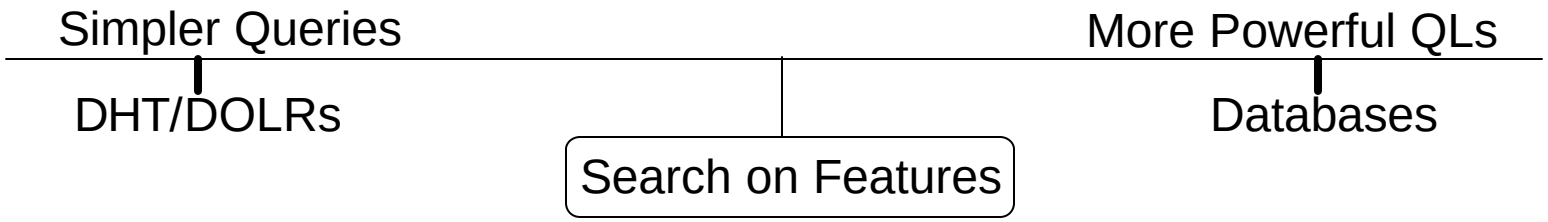
- Storage / query via structured P2P overlay networks
  - Large sparse ID space *N* (160 bits: 0 – 2^{160})
  - Nodes in overlay network have nodeIDs $\in N$
  - Given $k \in N$, overlay deterministically maps *k* to its ***root*** node (a live node in the network)
  - E.g. Chord, Pastry, **Tapestry**, Kademlia, Skipnet, etc…
- Decentralized Object Location and Routing (DOLR)
  - Objects identified by Globally Unique IDs (GUIDs) $\in N$
  - Decentralized directory service for endpoints/objects
    Route messages to ***nearest*** available endpoint
  - Object location with locality:
    routing stretch (overlay location / shortest distance) $\cong$ O(1)

DOLR on Tapestry Routing Mesh



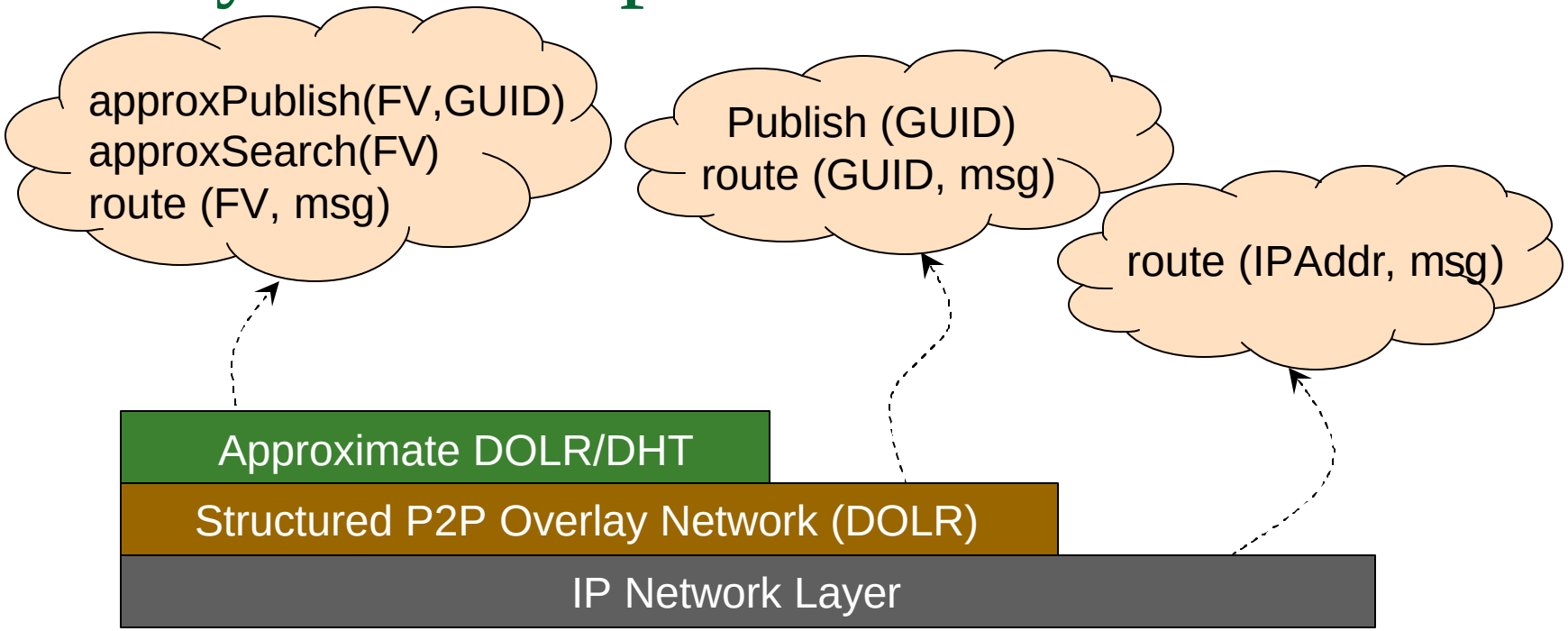
More Than Just Unique Identifiers

- Objects named by Globally Unique ID (GUID)
  - Application maps secondary characteristics to ID: versioning, modified replicas, app-specific info



- Simplify the search problem
  - ***out of* m *search fields, or "features," find objects matching at least* n *exactly***

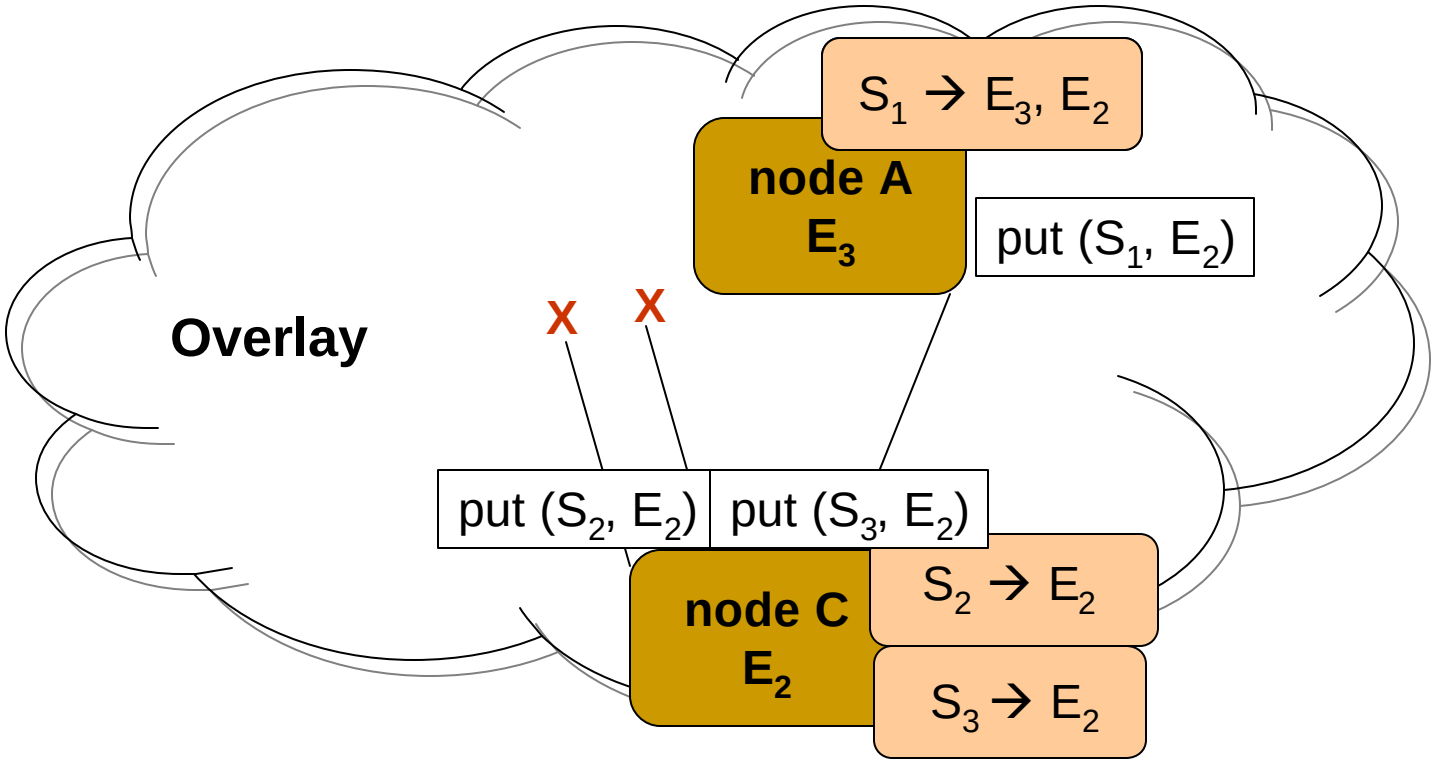
A Layered Perspective



ADOLR layer

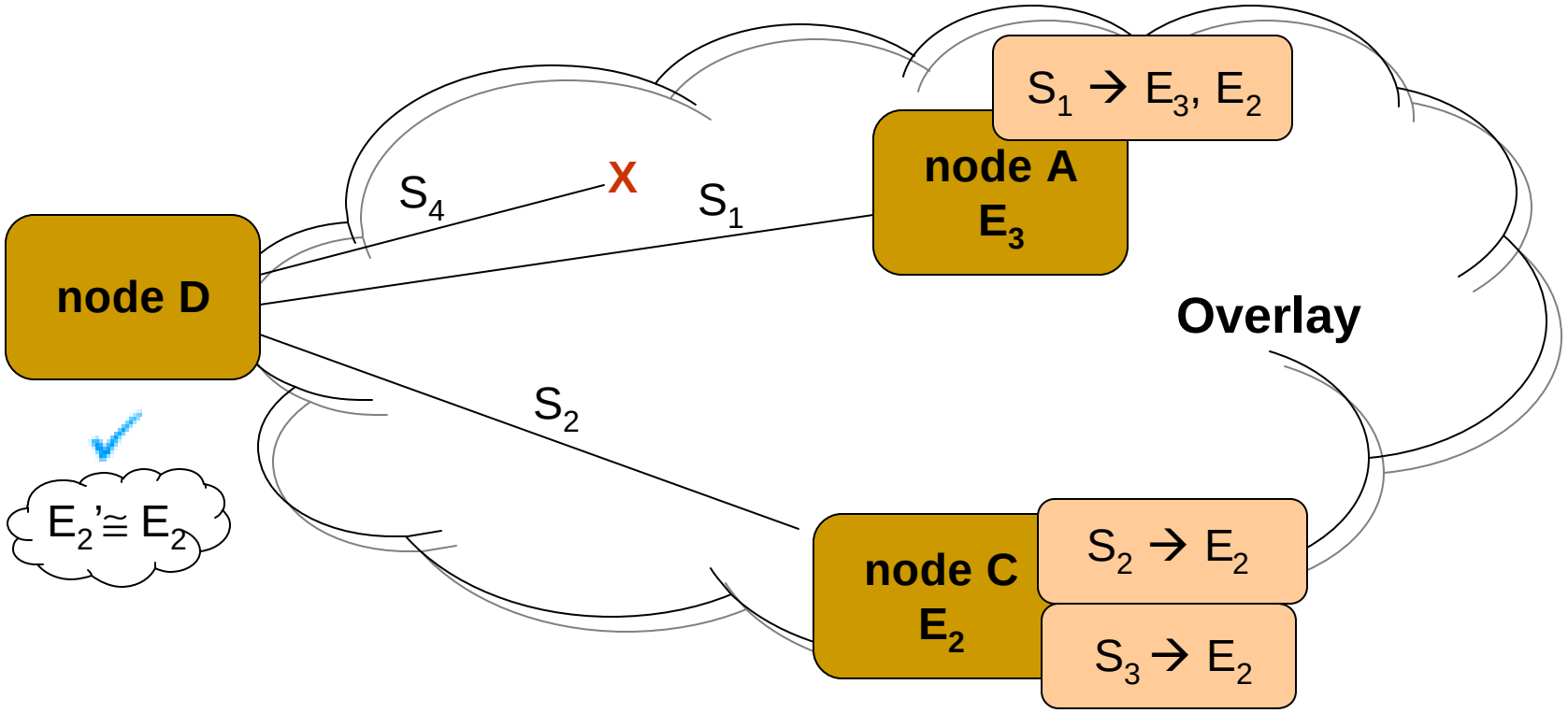
- Introduce naming mapping from feature vector to GUIDs
- Rely on overlay infrastructure for storage
- Abstraction of feature vectors as approximate names for object(s)

Marking a New Spam Message



- Signatures stored as inverted index (feature object) inside overlay
- User on C gets spam E_2, calculates signatures ***S***: {S_1, S_2, S_3}
- For each feature in ***S***, if feature object exists, add E_2
- If no feature object exists, create one locally and publish

Filtering New Emails for Spam

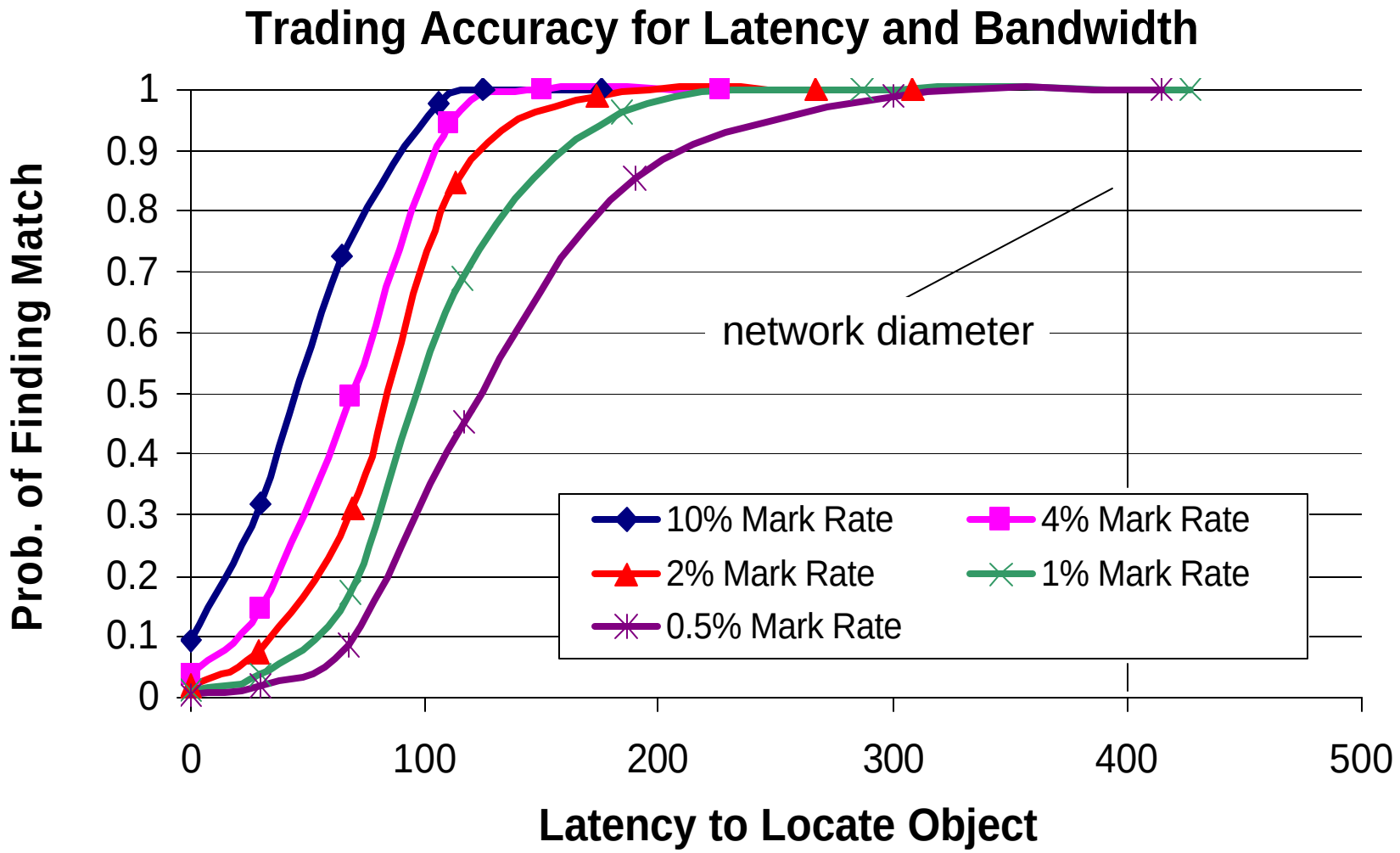


- User at node D receives new email E_2' with signatures $\{S_1, S_2, S_4\}$
- Queries overlay for signatures, retrieve matching GUIDs for each
- Threshold = 2/3, contact GUIDs that occur in 2 of 3 result sets
- Contact E_2 via overlay for any additional info (votes etc)

Constraining Bandwidth and Latency

- Need to constrain bandwidth and latency
  - Limit signature query to *h* overlay hops
  - Return null set if *h* hops reached without result
- Simulation on transit-stub topologies
  - 5K nodes, 4K overlay nodes, diameter = 400ms
  - Each spam message only reaches small group
    - For each message:
      % of users seen and marked = *mark rate*
  - Measure tradeoff between latency and success rate of locating known spam, for different mark rates

Simulation Result

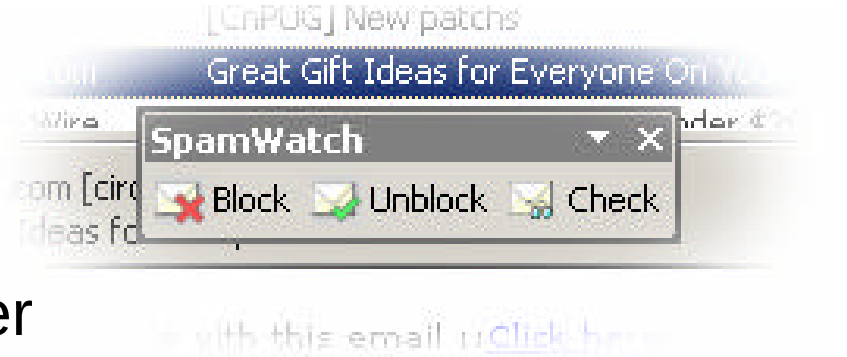


Feature-based Queries

- Approximate Text Addressing
  - Text objects: features → hashed signatures
  - Applications: plagiarism detection, replica management
- Other feature-based searching
  - Music similarity search
    - Extract musical characteristics
    - Signatures: {hash(field1=value1), hash(field2=value2)…}
    - E.g. Fourier transform values, # of wavetable entries
  - Image similarity search
    - Locate files with similar geometric properties, etc.

Finally…

- Status
  - ADOLR infrastructure implemented on Tapestry
  - **SpamWatch:** P2P spam filter implemented, including Microsoft Outlook plug-in
    Available for download:
    http://www.cs.berkeley.edu/~zf/spamwatch



- Tapestry
  http://www.cs.berkeley.edu/~ravenben/tapestry

- Thank you…Questions?