Detecting and Defending against Malicious Attacks in the iTrust Information Retrieval Network

Presented by Yung-Ting Chuang

Research conducted in collaboration with
Isai Michel Lombera, Louise E. Moser and P. M. Melliar-Smith
University of California, Santa Barbara
Supported in part by NSF Grant CNS 10-16103

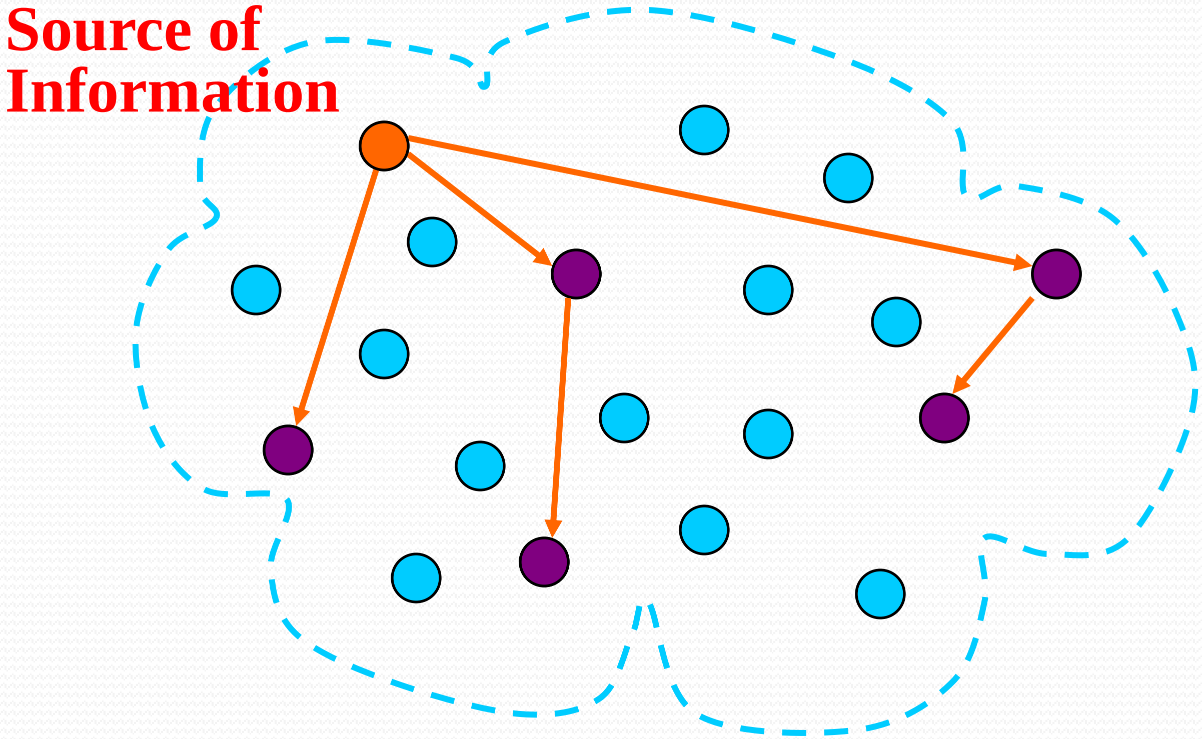
Overview

- Introduction
- Design of iTrust
- Implementation of iTrust
- Analytical Foundations
- Detecting Malicious Attacks
- Defending against Malicious Attacks
- Experimental Evaluation
- Related Work
- Conclusions and Future Work

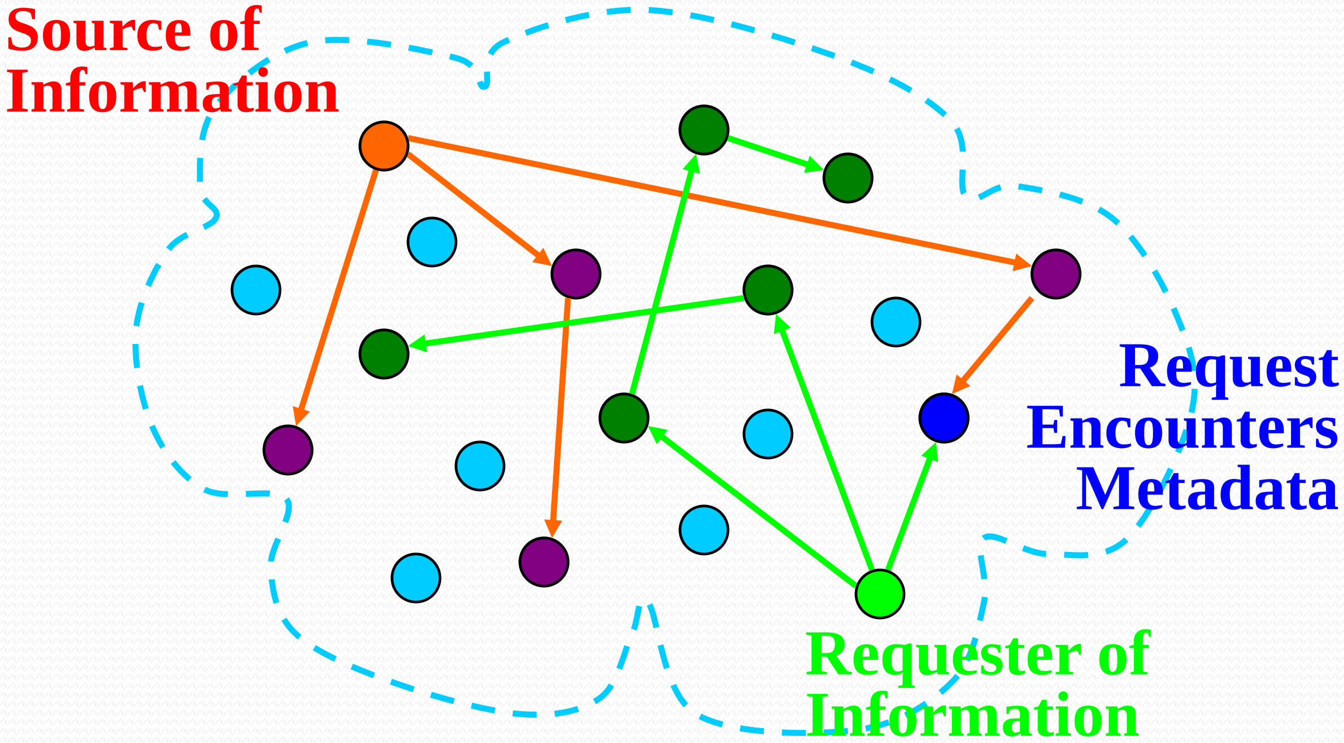
Introduction

- Today, we use centralized search engines on the Internet (Google, Yahoo!, Bing, etc)
  - Benefits
  - Drawbacks
- iTrust is desirable for individuals who fear censorship of information accessed on the Internet
- iTrust distributes metadata and requests to random participating nodes in the iTrust membership
  - Benefits
  - Drawbacks

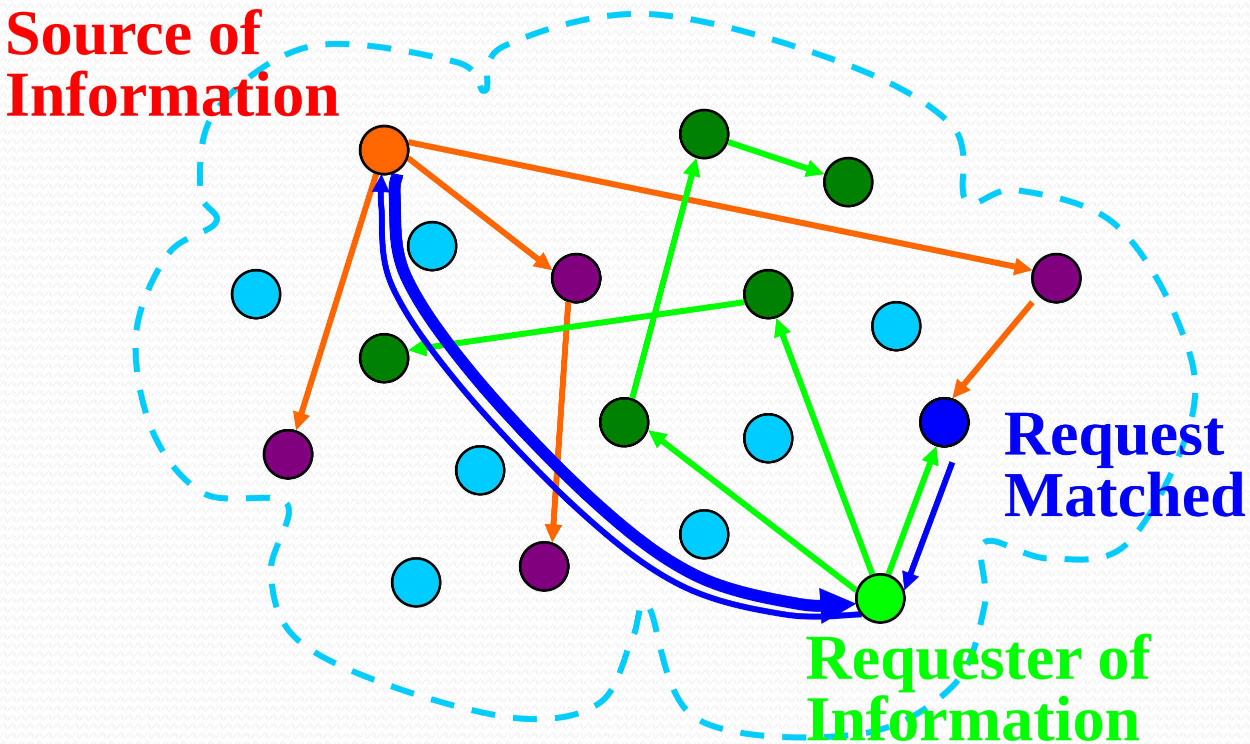
Design of iTrust



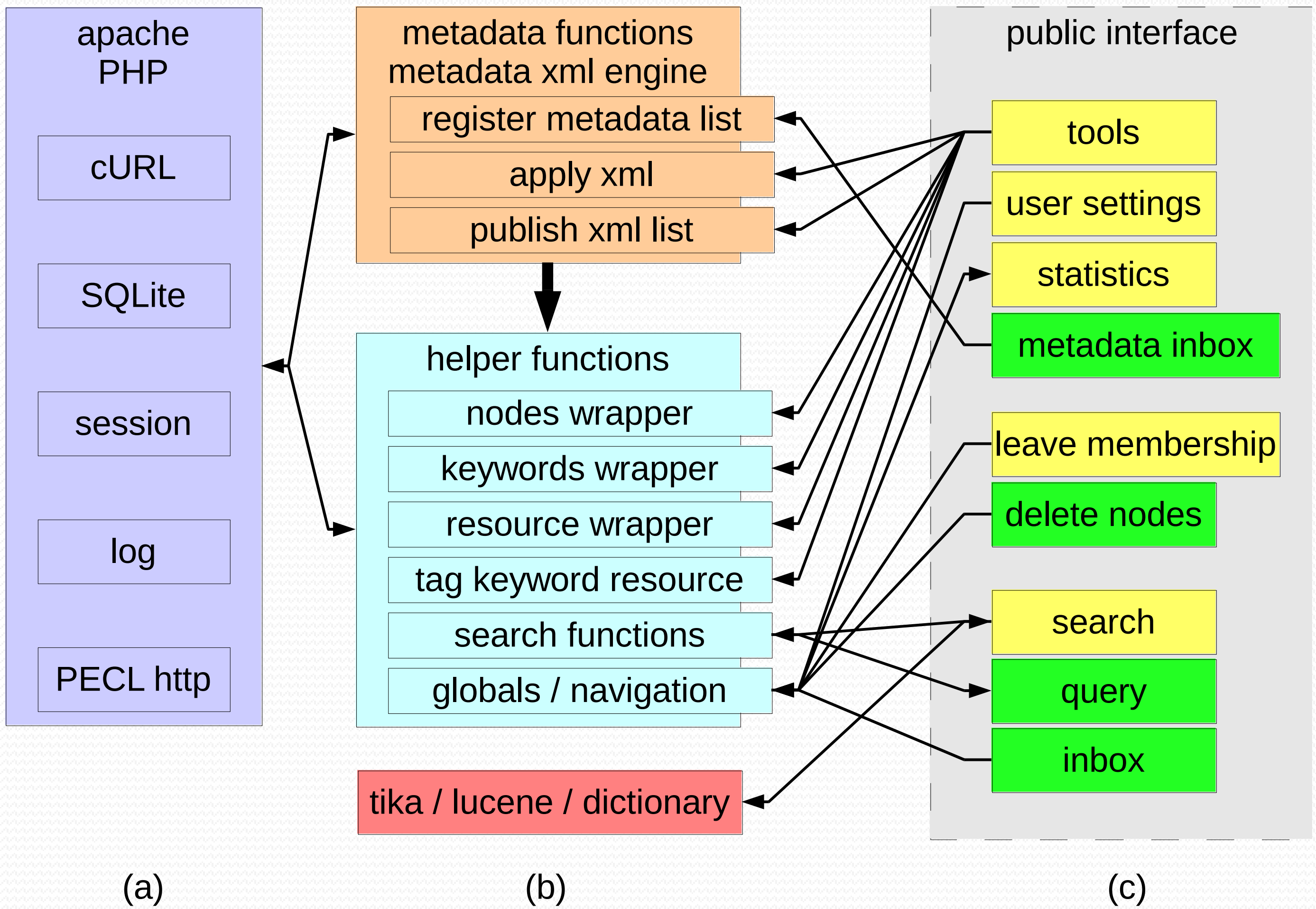
Design of iTrust



Design of iTrust



HTTP Implementation of iTrust



Analytical Foundations

- Assume
  - All nodes have the same membership set
  - Internet is reliable
  - All nodes has enough memory to store source documents

- Variables
  - Membership contains **n** participating nodes
  - **x** is the proportion of participating nodes that are operational
  - Metadata are distributed to **m** nodes
  - Requests are distributed to **r** nodes
  - **k** nodes report matches to a requesting node

iTrust Properties

- Probability p of ***k*** matches is

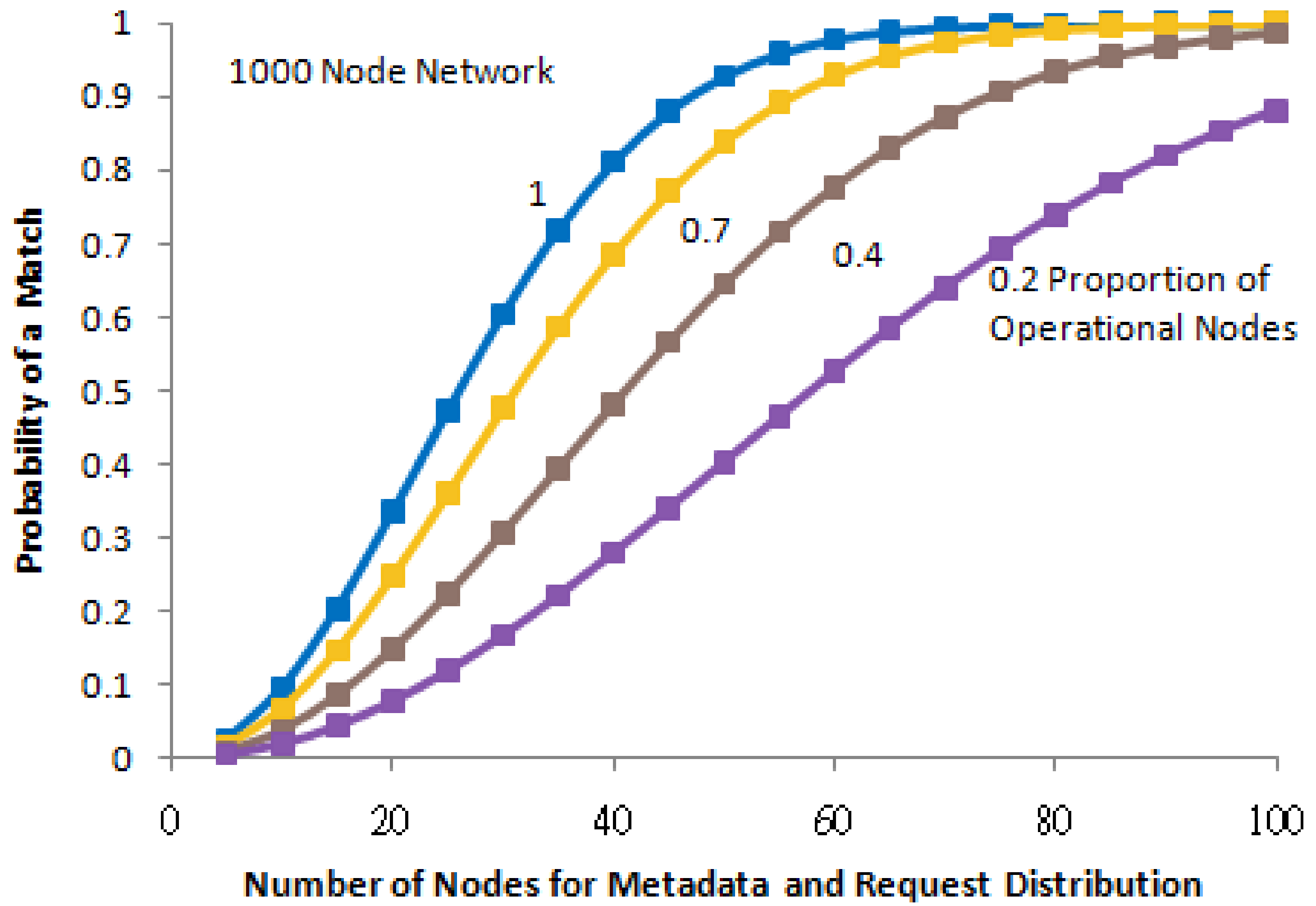
$$P(k) = \frac{\left(\frac{mx}{k}\frac{mx-1}{k-1}\cdots\frac{mx-k+1}{1}\right)\left(\frac{n-mx}{r-k}\frac{n-mx-1}{r-k-1}\cdots\frac{n-mx-r+k+1}{1}\right)}{\left(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1}\right)}$$

for $mx + r \leq n$ and $k \leq \min\{mx, r\}$

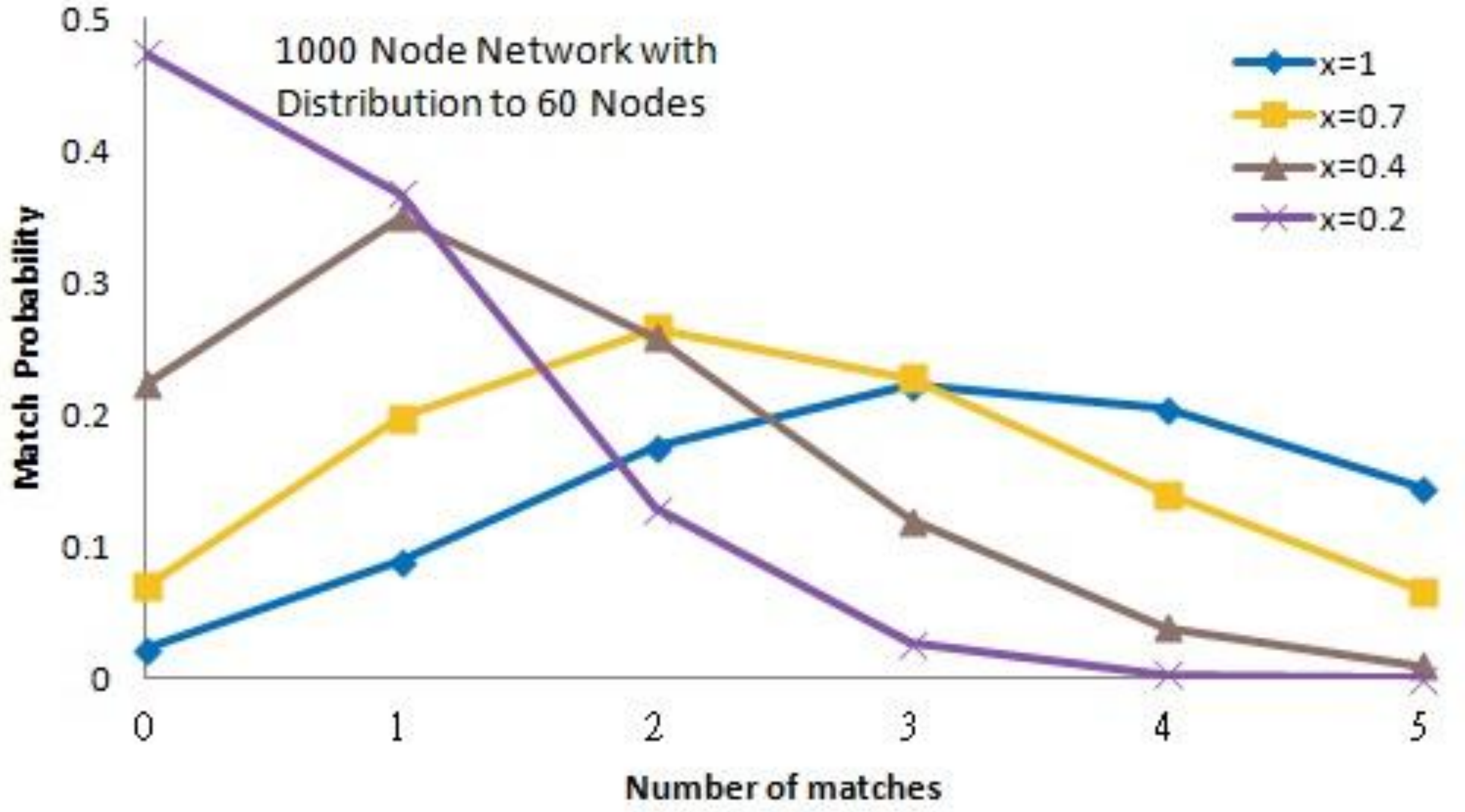
- Probability p of ***one or more*** matches is:

$$P(k \geq 1) = 1 - \frac{\left(\frac{n-mx}{r}\frac{n-mx-1}{r-1}\cdots\frac{n-mx-r+1}{1}\right)}{\left(\frac{n}{r}\frac{n-1}{r-1}\cdots\frac{n-r+1}{1}\right)} \quad \text{for } n \geq mx + r$$

Analytical Results



Detecting Malicious Attacks



Detecting Malicious Attacks

- Algorithm: Each requesting node
  - Makes requests and collect number of responses
  - Constructs empirical probabilities of number of matches
  - Computes analytical probabilities of number of matches
  - Uses Pearson's chi-squared goodness-of-fit test:

$$\chi^2 = \sum_{k=1}^{K} \frac{(o_k - e_k)^2}{e_k}$$

  - Determines x' from smallest χ^2 obtained in Pearson's test

Detecting Malicious Attacks

- Example: For 45 requests, we have:

  2 with 0 matches, 11 with1 match, 13 with 2 matches,

  10 with 3 matches, 7 with 4 matches, 2 with 5 matches
- Divide by 45 requests to get:

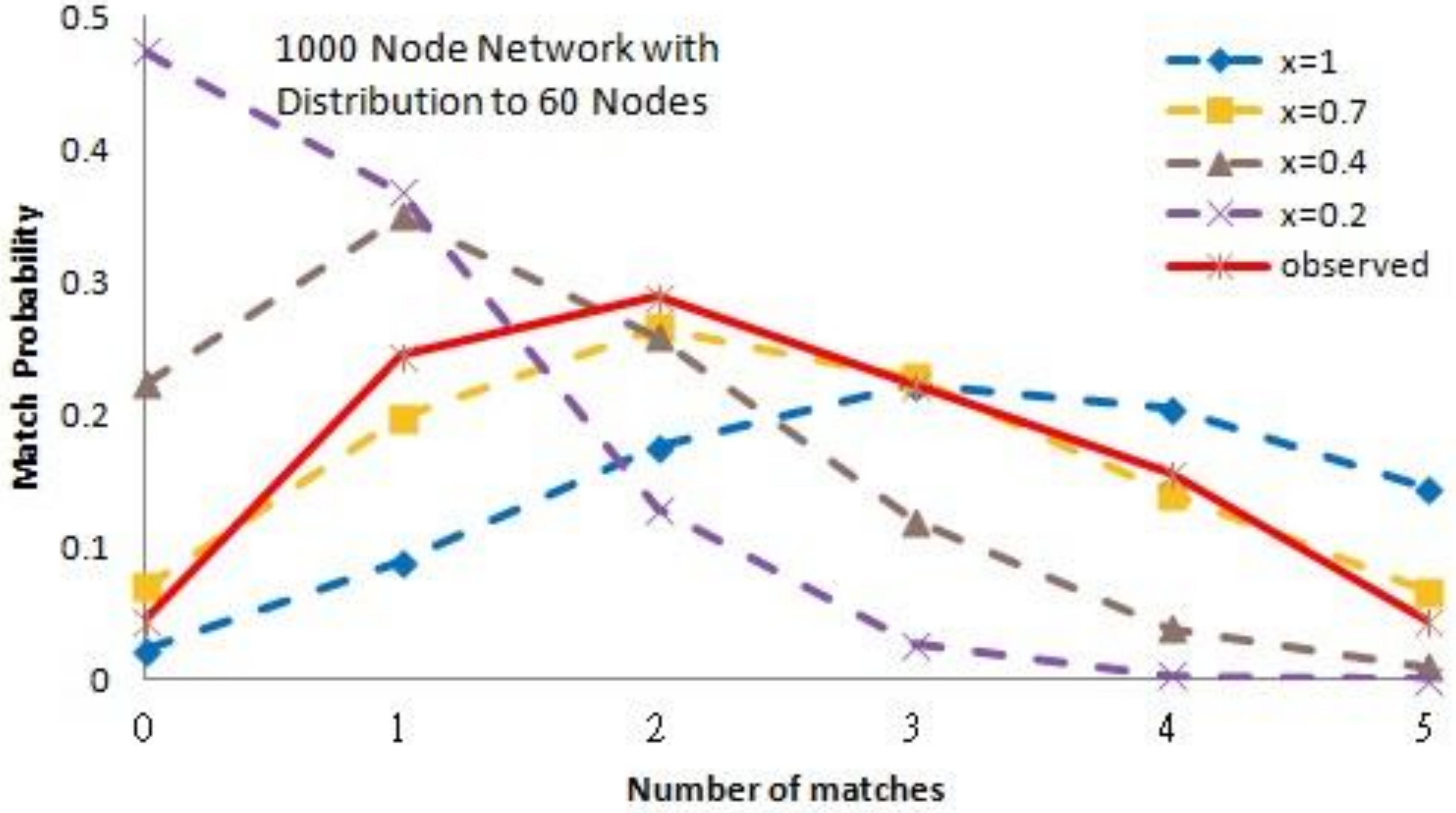
  $o_{k=1,2,3,4,5}$=(0.244,0.289,0.222,0.156,0.044)
- For x=1, calculate:

  $e_{k=1,2,3,4,5}$=(0.089,0.175,0.221,0.204,0.145)
- Apply Chi-Square test to get:

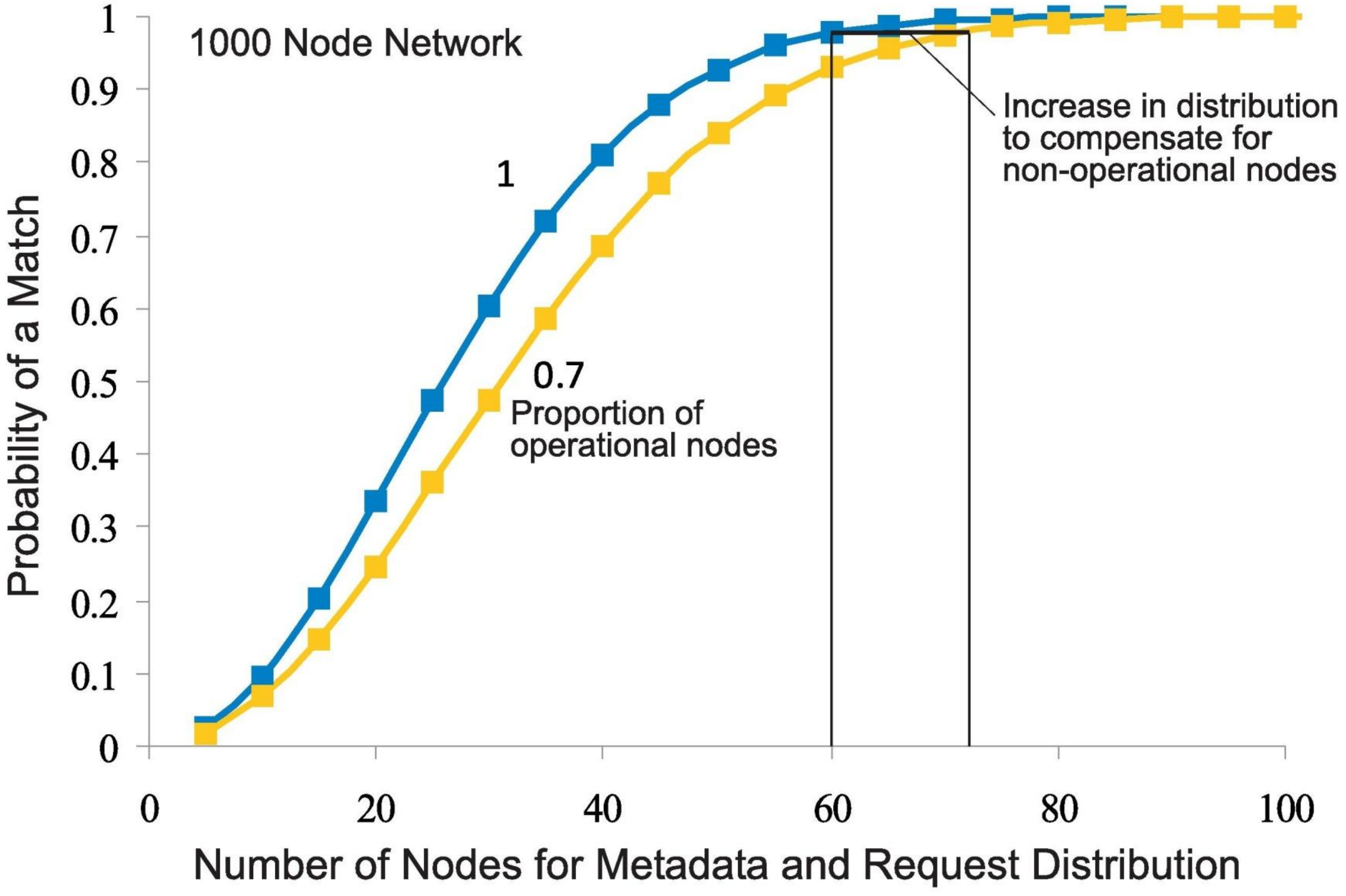
$$0.429 = \frac{(0.244-0.089)^2}{0.089} + \frac{(0.289-0.175)^2}{0.175} + \frac{(0.222-0.221)^2}{0.221} + \frac{(0.156-0.204)^2}{0.204} + \frac{(0.044-0.145)^2}{0.145}$$

- Repeat above steps for x=0.7, x=0.4, and x=0.2
- Compare all 4 Chi-Square values (0.429, 0.022, 0.603, 13.510)
- Smallest Chi-Square value=0.022, so the estimate is x'=0.7

Detecting Malicious Attacks



Defending Malicious Attacks



Defending Malicious Attacks

- Algorithm: Each node finds new values of m' and r':

Initialize n, x' $m' = m$ $r' = r$, y_0

Repeat

$m' = m' + 1;\ r' = r' + 1$

$$y' = 1 - \frac{(n - m'x')}{(n)} \frac{(n - m'x' - 1)}{(n - 1)} \cdots \frac{(n - m'x' - r' + 1)}{(n - r' + 1)}$$

until $y' > y_0$

return m', r'

Defending Malicious Attacks

- Example:

  Initially n=1000, x=1, r=m=60, y_o=0.978298

  Next detection algorithm estimates x'=0.7

  Then the defense algorithm calculates:

  x'=0.7, n=1000, r'=m'=60, y'=0.929725

  x'=0.7, n=1000, r'=m'=61, y'=0.935887

  x'=0.7, n=1000, r'=m'=62, y'=0.941604

  …

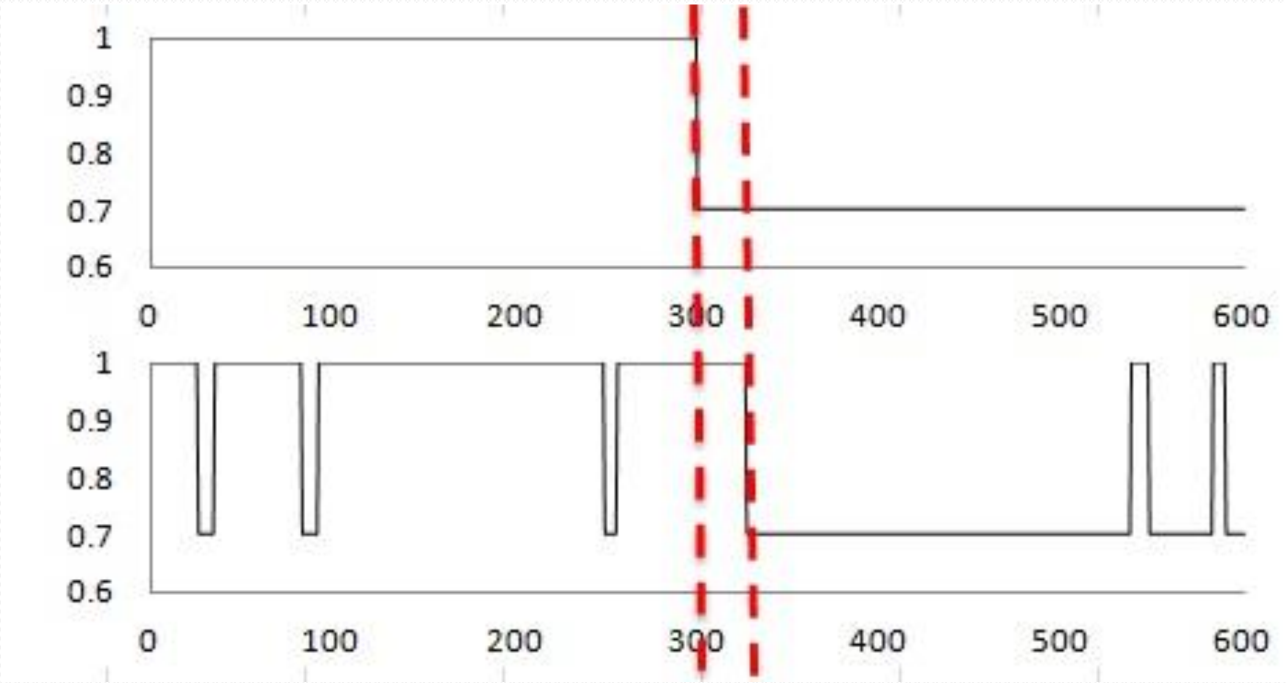
  x'=0.7, n=1000, r'=m'=71, y'=0.976623

  x'=0.7, n=1000, r'=m'=72, y'=0.979060 > y_o

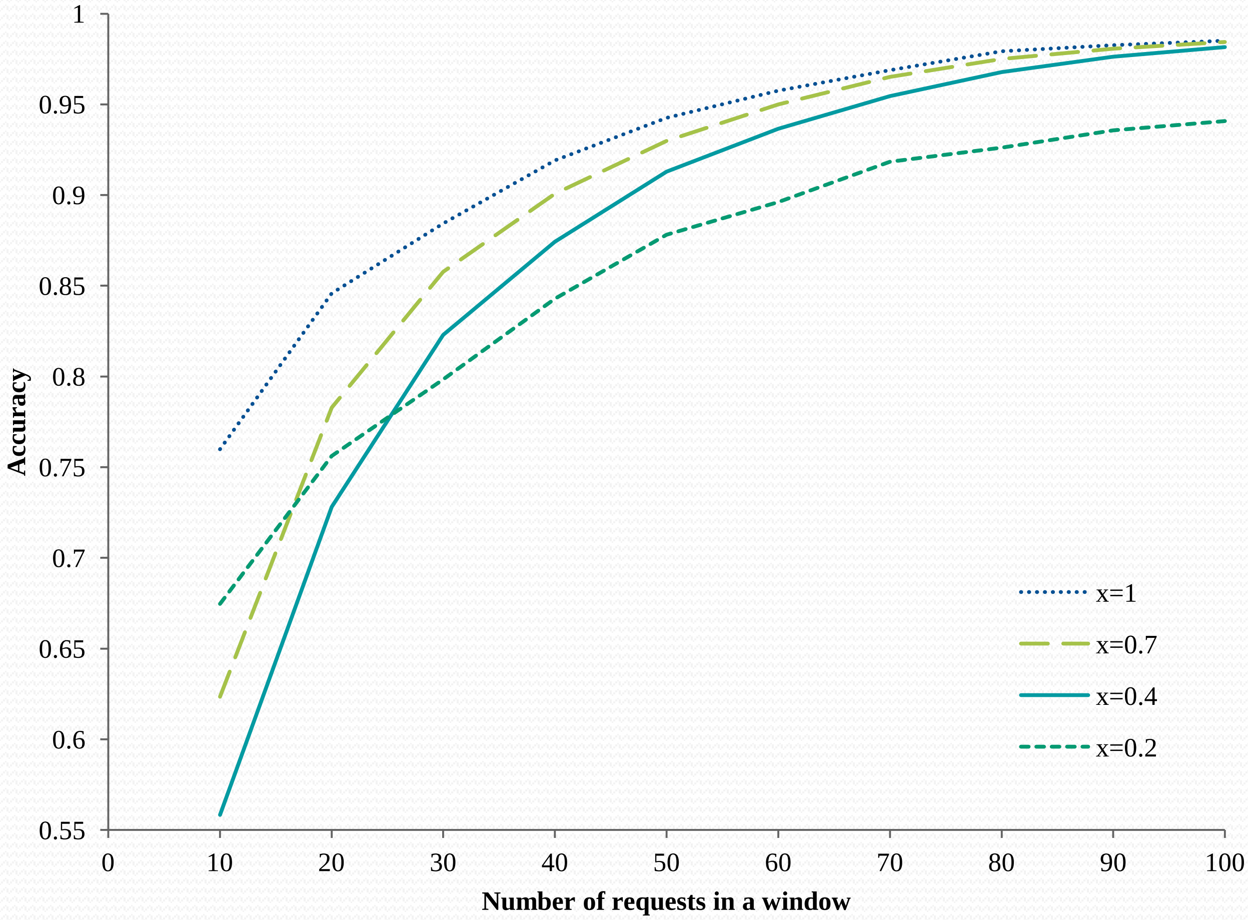
  The program stops and returns the new value r'=m'=72

Evaluation Metrics

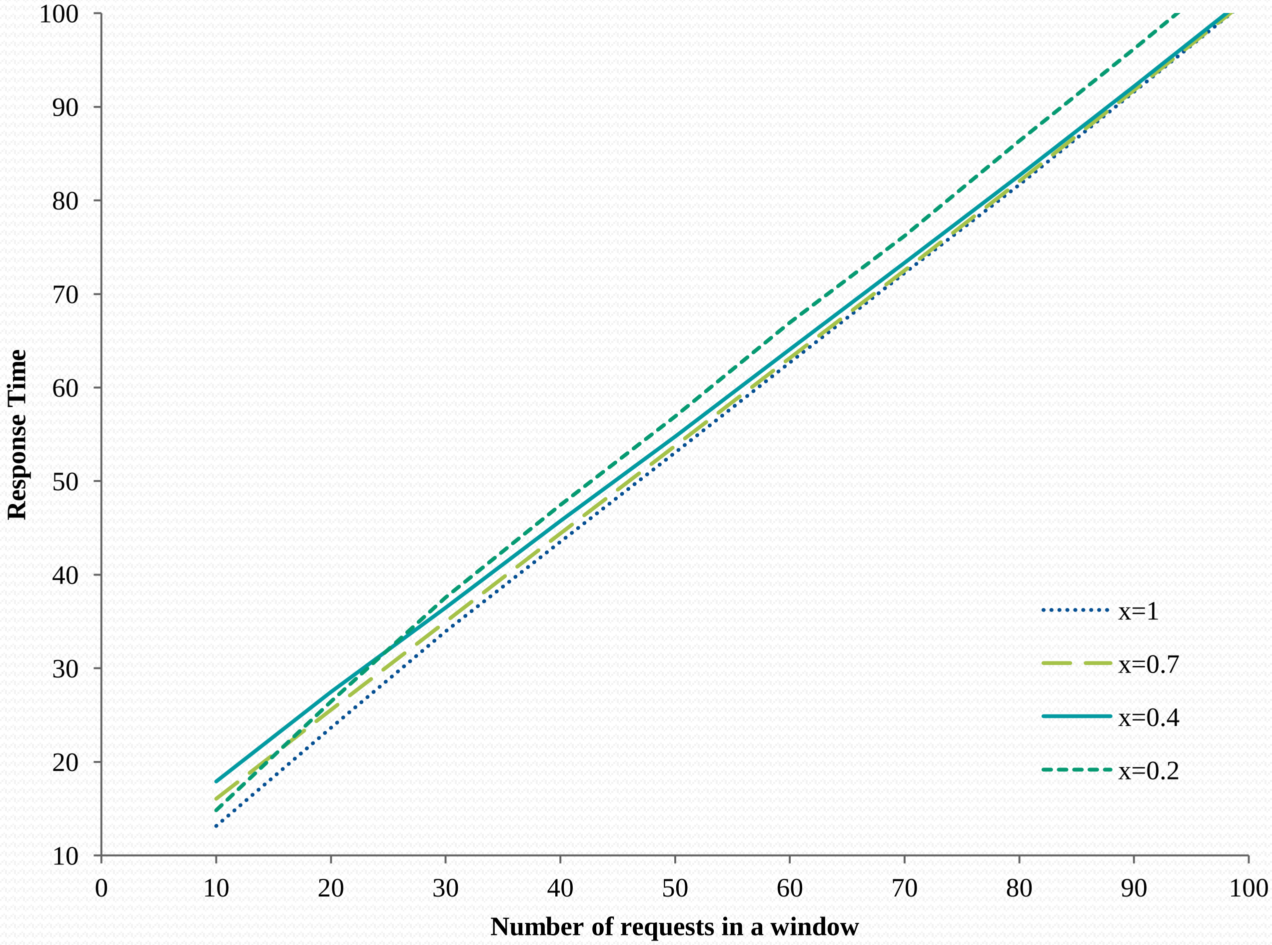
- Accuracy: $A(w) = \frac{\text{number of correct estimates of } x'}{\text{total number of estimates for a window size } w}$
- Response Time: $R(w) = \sum_{i=1}^{T} iwA(w)(1 - A(w))^{i-1}$



Experimental Evaluation (Accuracy)



Experimental Evaluation (Response Time)



Related Work

- Other unstructured publication, search, retrieval systems
  - Gnutella - uses flooding of requests
    - A node makes a copy of information when it receives the information it requested
    - Distributes requests up to maximum depth or time-to-live
  - Freenet – uses replication of information
    - Replicates information at a node, even if the node did not request it
    - Nodes that respond successfully to requests receive more metadata and more requests

Related Work

- Several recent publication, search, and retrieval systems are concerned with malicious attacks
  - Morselli – uses a feedback mechanism to determine if an object is replicated sufficiently
  - Jesi – uses gossip and blacklist mechanisms to identify malicious nodes, and focuses on hub attacks
  - Condie – uses local trust scores to find malicious peers that upload corrupt, inauthentic or misnamed content, and moves them to the edge of the network

Conclusions

- **iTrust is desirable for individuals who fear censorship of information accessed on the Internet**

- **We have presented novel statistical algorithms for detecting and defending against malicious attacks**
  - Detection algorithm estimates the proportion of nodes that are subverted or non-operational, based on the number of responses that a requesting node receives
  - Defensive adaptation algorithm determines the number of nodes to which the metadata and the requests must distributed to maintain the same probability of a match, as when all the nodes are operational

Future Work

- We plan to investigate other possible malicious attacks on iTrust and countermeasures to such attacks
- We are developing another implementation of iTrust based on SMS to operate on mobile phones
- We plan to make the iTrust source code, tools, and documentation freely available to all

Questions? Comments?

- Our iTrust Web Site
  - http://itrust.ece.ucsb.edu

- Contact information
  - Yung-Ting Chuang: ytchuang@ece.ucsb.edu
  - Isai: imichel@ece.ucsb.edu

- Our project is supported by NSF CNS 10-16193